
Learning with Relational Knowledge in the Context of Cognition, Quantum Computing, and Causality

Yunpu Ma 马鋆溥



München 2020

Learning with Relational Knowledge in the Context of Cognition, Quantum Computing, and Causality

Yunpu Ma 马鋆溥

Dissertation

an der Fakultät für Mathematik, Informatik und Statistik
der Ludwig-Maximilians-Universität
München

vorgelegt von

Yunpu Ma 马鋆溥

aus Tianjin, China

München, den 27.05.2020

Erstgutachter: Prof. Dr. Volker Tresp

Zweitgutachter: Prof. Dr. Christian Bauckhage

Drittgutachter: Prof. Dr. Evgeny Osipov

Tag der mündlichen Prüfung: 15.09.2020

Eidesstattliche Versicherung

(Siehe Promotionsordnung vom 12.07.11, § 8, Abs. 2 Pkt. .5.)

Hiermit erkläre ich an Eidesstatt, dass die Dissertation von mir selbstständig, ohne unerlaubte Beihilfe angefertigt ist.

Ma, Yunpu

.....
Name, Vorname

München, 27.05.2020

.....
Ort, Datum

Yunpu Ma

.....
Unterschrift Doktorand/in

Formular 3.2

Contents

Abstract	xi
Zusammenfassung	xiv
Acknowledgement	xviii
List of Publications and Declaration of Authorship	xxi
1 Introduction	1
1.1 Motivation	1
1.2 Modeling of Semantic Knowledge Graphs	5
1.2.1 Introduction of Semantic Knowledge Graphs	5
1.2.2 Relational Learning for Semantic Knowledge Graphs	7
1.3 Modeling of Episodic Knowledge Graphs	9
1.3.1 Relational Learning for Episodic Knowledge Graphs	9
1.3.2 Graph Hawkes Network for Modeling Episodic Knowledge Graphs .	10
1.4 Cognitive Perspectives of Knowledge Graphs	12
1.4.1 Knowledge Graphs and Cognitive Memories	12
1.4.2 Cognitive Architecture for Semantic Decoding	15
1.5 Holographic Reduced Representation and Holistic Representation	16
1.5.1 Holographic Reduced Representation	16
1.5.2 Quasi-Orthogonality	18
1.5.3 Holistic Representation	19
1.6 Variational Quantum Circuit for Knowledge Graph Embedding	21
1.6.1 Variational Quantum Circuit	21
1.6.2 Modeling Knowledge Graphs with Variational Quantum Circuit . .	24
1.7 Quantum Tensor SVD for Knowledge Graphs Inference	26

1.7.1	Classical Tensor Singular Value Decomposition	26
1.7.2	Sampling-based Quantum Algorithm for Knowledge Graphs Inference	30
1.8	Causal Inference under Interference	31
1.8.1	Introduction	31
1.8.2	GNN-based Causal Estimators	36
1.8.3	Intervention Optimization on Network	39
2	Embedding Models for Episodic Knowledge Graphs	41
2.1	Introduction	42
2.2	Model Descriptions	46
2.3	Experiments on Episodic Models	52
2.4	Semantic Memory from Episodic Memory with Marginalization	57
2.5	Conclusion	63
3	Holistic Representations for Memorization and Inference	69
3.1	Introduction	70
3.2	Representation Learning	71
3.3	Derivation of ϵ -Orthogonality	72
3.3.1	ϵ -Orthogonality for a Gaussian Distribution	72
3.3.2	ϵ -Orthogonality for a Cauchy Distribution	73
3.4	Holistic Representations for KGs	74
3.4.1	HRR Model	74
3.4.2	Holistic Model	74
3.4.3	Experiments on Memorization	75
3.4.4	Correlation versus Convolution	75
3.4.5	Hyper-parameter ξ	76
3.5	Inference on KG	77
3.5.1	Inference via Holistic Representation	77
3.5.2	Inference on New Entities	78
3.6	Conclusion	78
3.7	Appendix	81
4	Variational Quantum Circuit Model for Knowledge Graph Embedding	91
4.1	Introduction	92
4.2	Representation Learnings on Knowledge Graphs	92

4.2.1	Knowledge Graphs	92
4.2.2	Representation Learning	93
4.3	Quantum Circuit Models	94
4.4	Circuit Models for Knowledge Graphs	94
4.4.1	Quantum Circuit Embedding	94
4.4.2	Loss Function and Training	96
4.4.3	Fully Parameterized Quantum Circuit Embedding	96
4.5	Experiments	97
4.5.1	Datasets and Evaluation	97
4.5.2	Regularizations	98
4.5.3	T-SNE	100
4.6	Accelerated Inference	100
4.7	Conclusion and Outlook	102
4.8	Appendix	102
5	Quantum Machine Learning Algorithm for Knowledge Graphs	105
5.1	Introduction	106
5.1.1	Related Work	107
5.2	Tensor Singular Value Decomposition	108
5.3	Quantum Machine Learning Algorithm for Knowledge Graphs	111
5.3.1	Quantum Mechanics	111
5.3.2	Quantum Tensor Singular Value Decomposition	112
5.4	Experiments with Classical Tensor SVD	115
5.5	Conclusion	116
5.6	Appendix	117
6	Causal Inference under Networked Interference and Intervention Policy Enhancement	131
6.1	Introduction	132
6.1.1	Related Work	133
6.1.2	Notations and Previous Approaches	133
6.2	GNN-based Causal Estimators	134
6.2.1	Structural Equation Model	134
6.2.2	Distribution Discrepancy Penalty	135
6.2.3	Graph Neural Networks	135

6.2.4	GNN-based Causal Estimators	136
6.3	Intervention Policy on Graph	137
6.4	Experiments	141
6.4.1	Datasets	141
6.4.2	Results of Causal Estimators	142
6.4.3	Results on Improved Intervention Policy	145
6.5	Conclusion	147
6.6	Appendix	150
7	Conclusion	171

Abstract

Relational learning algorithms utilize structured relational data and underlying structure to extract useful information. Examples of relational data are social networks that can be abstracted as graphs with undirected edges or triple-oriented knowledge graphs as directed graphs with labeled edges. In recent years, large-scale triple-oriented knowledge graphs have become an essential approach for knowledge representation and reasoning. They are widely used in artificial intelligence systems, such as question answering systems, recommendation systems, etc. One well-known example is IBM's cognitive computing platform, IBM Watson, where the knowledge graph is a core component. Another example is the world's largest structured relational database of human knowledge, Google's knowledge graph. Knowledge graphs are also applied in enterprises as knowledge management and process monitoring tools. For example, inside Siemens, a knowledge graph for gas turbines is applied to realize knowledge-based maintenance.

Triple-oriented knowledge graphs consist of semantic triples, which are interlinked entities describing the facts and human knowledge of the world, e.g., (*California, locatedIn, USA*). Triple-oriented knowledge graphs are static and represent the human knowledge of the world at a specific timestamp. However, commonly, the world is changing, as well as the human knowledge of it. For example, a healthy person becomes diagnosed with a disease, or a new president is inaugurated. Hence, semantic triples can be easily generalized to episodic quadruples by incorporating timestamps, and episodic quadruples constitute an episodic knowledge graph describing the evolving knowledge and changing facts of a dynamic world. Modeling episodic knowledge graphs using tensor methods for knowledge inference is one of the subjects studied in the present dissertation.

Triple-oriented semantic knowledge graphs and quadruple-oriented episodic knowledge graphs are supposed to be the most straightforward and human-understandable form of static and evolving knowledge, respectively. Hence, it is hypothesized that semantic and episodic knowledge graphs are technical realizations of the brain's declarative memories in

artificial intelligence systems. As the first contribution, in this dissertation, we closely investigate the corresponding relation between semantic memory and the semantic knowledge graph, as well as episodic memory and the episodic knowledge graph. The interdependency between semantic and episodic memory suggests that their technical realizations, the semantic and episodic knowledge graphs, are also not mutually independent and support one another. We can realize a mapping from the episodic knowledge to the semantic knowledge, or, from the cognitive perspective, a transition from the episodic memory to the semantic memory, via marginalization of timestamps. Cognitive aspects of different knowledge graphs are wholly original and open new research directions. For example, previous studies showing that semantic knowledge graph can improve visual relationship detection in images suggest that there could be a deeper connection between perception and semantic memory.

Knowledge graphs can be constructed by gathering and merging information from unstructured texts of different resources. Hence, the size of knowledge graphs, including the number of semantic triples and the number of distinguishable entities, is rapidly growing. For example, the size of Google’s knowledge graph grows rapidly after launch, and currently, it contains more than 570 million distinguishable entities and nearly 100 billion semantic triples. However, the speed of performing knowledge inference on knowledge graphs is greatly influenced by the massive number of triple facts and entities. Considering that all the current learning algorithms on knowledge graphs are classical algorithms and implemented on classical computational resources, as the second contribution of this dissertation, we develop first quantum learning algorithms that can potentially speedup the knowledge inference on knowledge graphs.

We propose two quantum machine learning algorithms: a trainable quantum circuit-based method, and a quantum tensor singular value decomposition method. Both show different speedups on inference in the knowledge graph. In particular, the quantum circuit-based method applies variational quantum circuits to obtain quantum representations of entities and evaluates the score functions of semantic triples via unitary circuit evolution. We prove that the resulting method shows a quadratic speedup with respect to the number of entities and could, in principle, be tested on noisy intermediate-scale quantum devices. We further investigate the performance of the circuit-based method via a quantum variational simulator. While the quantum circuit-based method is parametric and trainable, the quantum tensor singular value decomposition method performs knowledge inference via sampling, showing an exponential acceleration with respect to the number

of entities. Despite a dramatically reduced inference time, the quantum sampling-based method cannot easily be implemented on current devices, since its quantum subroutines, e.g., quantum random access memory and quantum phase estimation, are nontrivial and require error corrects. As one of the theoretical contributions of this dissertation, we prove the plausibility of the sampling-based quantum algorithm theoretically and investigate its performance by implementing classical tensor singular value decomposition on knowledge graphs.

In the last part of this dissertation, we study causal effects on relational data. Causal inference on relational data is a relatively new research area. Hence, as a first step, instead of studying causal effects on knowledge graphs, we investigate the causal effects in social networks and user-item networks. Commonly, consistency and lack of interference are underlying assumptions in studies of causal inference. However, the interference-free assumption becomes improper in the social network setting. For example, the treatment response of a unit in the social network can be affected by the treatment assignments or responses of its neighboring units. Hence, the treatment response of a unit is a superposition of individual treatment effect and peer effect. As the third contribution of this dissertation, we propose the first causal estimators to estimate superimposed causal effects on networks using different graph neural networks. After obtaining graph neural network-based causal estimators, we employ an optimal policy network for treatment assignment on the network to maximize the network's total welfare.

Zusammenfassung

Relationale Lernalgorithmen verwenden strukturierte relationale Daten und die zugrunde liegende Struktur, um nützliche Informationen zu extrahieren. Beispiele für relationale Daten sind soziale Netzwerke, die als Graphen mit ungerichteten Kanten betrachtet werden können, oder Tripel-orientierte Wissensgraphen, die als gerichtete Graphen mit beschrifteten Kanten betrachtet werden können. In den letzten Jahren sind Tripel-orientierte Wissensgraphen zu einem wesentlichen Ansatz für die Präsentation von Wissen geworden. Sie werden häufig in Systemen der künstlichen Intelligenz verwendet, wie z.B. in Fragenbeantwortungssystemen, Empfehlungssystemen usw. Ein bekanntes Beispiel ist die IBM kognitive Computing-Plattform, IBM Watson, bei der ein Wissensgraph eine Kernkomponente ist. Ein weiteres Beispiel ist die weltweit größte relationale Datenbank menschlichen Wissens, der Wissensgraph von Google. Wissensgraphen werden auch in Unternehmen als Werkzeuge für Wissensmanagement und Prozessüberwachung eingesetzt. So wird beispielsweise innerhalb von Siemens ein Wissensgraph für Gasturbinen für bessere Wartungszyklen angewendet.

Tripel-orientierte Wissensgraphen bestehen aus semantischen Tripeln, die miteinander verbindenden Entitäten darstellen. Semantische Tripel beschreiben die Fakten und das menschliche Wissen der Welt, z.B. (Kalifornien, Lokalisiert In, USA). Tripel-orientierte Wissensgraphen sind statisch und stellen das menschliche Wissen zu einem bestimmten Zeitstempel dar. Im Allgemeinen verändert sich jedoch die Welt, ebenso wie das menschliche Wissen darüber. Zum Beispiel wird bei einer gesunden Person eine Krankheit diagnostiziert oder ein neuer Präsident wird vereidigt. Daher können semantische Tripel leicht auf episodische Quadrupel verallgemeinert werden, indem Zeitstempel integriert werden. Episodische Quadrupel bilden episodische Wissensgraphen, die das sich entwickelnde Wissen und die sich verändernden Fakten der dynamischen Welt beschreiben. Die Modellierung episodischer Wissensgraphen mit Tensor Zerlegung zur Wissensinferenz ist eines der in dieser vorliegenden Arbeit untersuchten Themen.

Tripel-orientierte semantische und Quadrupel-orientierte episodische Wissensgraphen sollen die direkteste und menschenverständlichste Form von statistischem bzw. sich entwickelndem Wissen sein. Daher wird die Hypothese aufgestellt, dass semantische und episodische Wissensgraphen die technischen Realisierungen der deklarativen Gedächtnisse des Gehirns in künstlichen Intelligenzsystemen sind. Als erster Beitrag untersuchen wir in dieser Dissertation die entsprechende Beziehung zwischen dem semantischen Gedächtnis und dem semantischen Wissensgraphen sowie Beziehung zwischen dem episodischen Gedächtnis und dem episodischen Wissensgraphen. Die Interdependenz zwischen semantischem und episodischem Gedächtnis zeigt, dass ihre technischen Realisierungen, die semantischen und episodischen Wissensgraphen, auch nicht voneinander unabhängig sind und sich gegenseitig unterstützen. Wir können eine Abbildung vom episodischen Wissen zum semantischen Wissen oder, aus der kognitiven Perspektive, einen Übergang vom episodischen Gedächtnis zum semantischen Gedächtnis durch Marginalisierung von Zeitdimension realisieren. Kognitive Perspektiven verschiedener Wissensgraphen sind völlig original und eröffnen neue Forschungsrichtungen. Frühere Studien, die zeigen, dass semantische Wissensgraphen die visuelle Erkennung von Beziehungen in Bildern verbessern können, deuten beispielsweise darauf hin, dass ein tieferer Zusammenhang zwischen Wahrnehmung und semantischem Gedächtnis besteht.

Wissensgraphen können aufgebaut werden, indem Informationen aus den unstrukturierten Texten verschiedener Ressourcen gesammelt und extrahiert werden. Daher wächst die Größe der Wissensgraphen, einschließlich der Anzahl der semantischen Tupel und der Anzahl der Entitäten, schnell. Zum Beispiel nimmt die Größe von Googles Wissensgraphen schnell zu, und derzeit enthält er mehr als 70 Millionen unterscheidbare Entitäten und fast 100 Milliarden semantische Tripel. Die Geschwindigkeit der Durchführung von Wissensinferenz in Wissensgraphen wird jedoch stark durch die enorme Anzahl semantischer Fakten und Entitäten beeinflusst. In Anbetracht der Tatsache, dass alle aktuellen Lernalgorithmen für Wissensgraphen klassische Algorithmen sind und auf klassischen Rechenressourcen implementiert sind, entwickeln wir als zweiten Beitrag dieser Dissertation die ersten Quantenalgorithmen, die möglicherweise die Wissensinferenz in Wissensgraphen beschleunigen können.

Wir schlagen zwei Quantenalgorithmen vor: eine trainierbare Quantenschaltkreis-basierte Methode und eine Quantentensor-Singulärwert-Zerlegungsmethode. Beide Quantenalgorithmen zeigen unterschiedliche Beschleunigungen der Wissensinferenz in Wissensgraphen. Insbesondere wendet die Quantenschaltkreis-basierte Methode variationellen Quantenschaltkreise

an, um die Quantendarstellungen von Entitäten zu erhalten. Die Score-Funktionen semantischer Tripel werden über die unitäre Evolution der Quantenschaltung berechnet. Wir haben bewiesen, dass die Quantenschaltkreis-basierte Methode eine quadratische Beschleunigung in Bezug auf die Anzahl der Entitäten bei der Wissensinferenz aufweist. Wir untersuchten weiterhin die Leistungen der schaltungsbasierten Methode mit Hilfe eines Quantensimulators.

Während das quantenschaltungs-basierte Verfahren parametrisiert und trainierbar ist, führt die Quantentensor-Singulärwert-Zerlegungsmethode die Wissensinferenz durch das Sampling durch und zeigt eine exponentielle Beschleunigung in Bezug auf die Anzahl der Entitäten. Trotz einer drastisch verkürzten Inferenzzeit lässt sich diese Quantensampling-basierte Methode nicht leicht auf gegenwärtigen Rechner implementieren, da die benötigten Quantensubroutinen, z.B. der Quantenzugriffsspeicher und die Quantenphasenschätzung, nicht trivial sind und Quantenfehlerkorrekturen erfordern. Als einer der theoretischen Beiträge dieser Arbeit beweisen wir die Plausibilität des Sampling-basierten Quantenalgorithmus und untersuchen seine Leistung, indem wir klassische Tensor-Singulärwertzerlegung von Wissensgraphen implementieren.

Im letzten Teil dieser Dissertation untersuchen wir kausale Effekte auf relationale Daten. Kausale Inferenz auf relationale Daten ist ein relativ neues Forschungsgebiet. Daher untersuchen wir in einem ersten Schritt die kausalen Auswirkungen in sozialen Netzwerken und Benutzer-Item-Netzwerken. Im Allgemeinen sind die Konsistenz und das Fehlen von Interferenz die zugrunde liegenden Annahmen in Studien der kausalen Inferenz. Die interferenzfreie Annahme wird jedoch in der Einstellung der sozialen Netzwerke unrealistisch. Beispielsweise kann die Behandlungsreaktion einer Einheit im sozialen Netzwerk durch die Behandlungszuweisungen und die Behandlungsreaktionen der benachbarten Einheiten beeinflusst werden. Daher stellt die Behandlungsreaktion einer Einheit eine Überlagerung des individuellen Behandlungseffekts und Peer-Effekts dar. Als dritten Beitrag dieser Dissertation schlagen wir die ersten Kausalschätzer vor, die verschiedene Graph neuronaler Netzwerke verwenden, um überlagerte Kausaleffekte im Netzwerk zu schätzen. Nach dem Erhalt von Graph neuronalen Netzwerke-basierten Kausalschätzern, setzen wir eine optimale Strategie für die Behandlungszuweisung im Netzwerk ein, um das Gesamtwohl des Netzwerks zu maximieren.

Acknowledgement

This dissertation is based on the scientific outputs I 马鋆溥 delivered as a doctoral student of Ludwig Maximilian University of Munich and a research scientist of the Machine Learning Group at Siemens Corporate Technology. My research journey and the completion of this dissertation would not have been possible without the support and contribution of many inspiring people.

First, I would like to express my sincere gratitude to my doctoral supervisor, Doktorvater in German, Professor Volker Tresp for his extraordinary support of my doctoral research, for his constructive advice, patience, and his invaluable insight into research. I am also grateful to Volker, who provided me the opportunity to study and do research in machine learning and artificial intelligence, although I graduated from physics. He offered me a chance to participate in the Cognitive Deep Learning funded by Siemens Corporate Technology. As an extraordinary mentor, he introduced me to knowledge graphs. He revealed to me the deep and most insightful relationship between knowledge graphs and cognitive memory functions, which becomes one primary subject studied in this dissertation.

I have been very fortunate that Volker encouraged me to innovate and explore new directions and provided constructive feedback, such that I can make contributions to the areas of quantum machine learning and causal inference, which became the other topics of this dissertation. Many thanks to Volker for the detailed comments on all my publications and the draft of this dissertation. I also wish to thank Volker and Siemens CT, who generously funded me to participate in different academic workshops and conferences held in Austria, Spain, the USA, and Canada and helped me building academic connections. Special thanks go to Prof. Dr. Christian Bauckhage and Prof. Dr. Evgeny Osipov for agreeing to be the external examiners of my dissertation.

I am also profoundly grateful for the contribution of my co-authors and enjoy the collaboration with them. Dr. Stephan Baier brought me to his Visual Relation Detection project. Marcel Hildebrandt and I formed an exciting idea of debating on knowledge graphs.

I enjoy the collaboration with Zhiliang Wu for his project of causal effect estimation. Zhen Han and I collaborate on several papers of temporal knowledge graphs and keep pushing this research topic further. Dr. Yuyi Wang introduced me to fantastic research areas of theoretical computer science, such as distributed computing and blockchain. Many thanks to all other great colleagues and collaborators, especially Erik Daxberger, Dr. Yinchong Yang, Dr. Denis Krompass, Dr. Sigurd Spieckermann, Dr. Liming Zhao, etc. I'd like to acknowledge the assistance provided by Siemens CT and Dr. Ulli Waltinger for patenting some of my works.

To my dear Family, thank you for encouraging me and supporting me to chase my dreams, especially my mom Wang Chunyan 王春燕, my aunt Ma Jun 马军, and my uncle Wang Peiwu 王培武. I am also lucky to have lovely friends who helped me through the journey, especially Ye Jiang, Dr. Kun Qian, Yuanyuan Yang, and Thomas Lares, etc.

List of Publications and Declaration of Authorship

- Yunpu Ma, Volker Tresp, and Erik Daxberger. Embedding Models for Episodic Knowledge Graphs. Journal of Web Semantics, Special Issue on Representation Learning, volume 59 (2019), pages 100490. Elsevier. doi:10.2139/ssrn.3319790

I conceived of the presented idea, performed all computations and evaluations. I wrote the initial manuscript and did most of the subsequent revisions. Erik Daxberger helped me drawing the illustrative figure. I regularly discussed this work with my supervisor Prof. Volker Tresp, who supervised the finds of this work. All co-authors discussed the results periodically and assisted in polishing the final manuscript.

Chapter 2 presents this work.

- Yunpu Ma, Marcel Hildebrandt, Volker Tresp, Stephan Baier. Holistic Representations for Memorization and Inference. In proceedings of the Conference on Uncertainty in Artificial Intelligence (UAI), pages 403-413. Monterey, California, USA, August 6 - 10, 2018. Archived at <http://auai.org/uai2018/proceedings/papers/163.pdf>, without doi.

I conceived of the presented idea, performed all computations, and developed the whole theory. I wrote the initial manuscript and did most of the subsequent revisions. Marcel Hildebrandt pointed out a mistake in the proof of the initial draft. I regularly discussed this work with my supervisor Prof. Volker Tresp, who supervised the finds of this work. All co-authors discussed the results periodically and assisted in polishing the final manuscript.

Chapter 3 presents this work.

- Yunpu Ma, Volker Tresp, Liming Zhao, and Yuyi Wang. Variational Quantum Circuit Model for Knowledge Graphs Embedding. Advanced Quantum Technologies, 2(7-8), 1800078. First published: February 12, 2019. Wiley. doi:10.1002/qute.201800078

I conceived of the presented idea, performed all computations and evaluations. I wrote the initial manuscript and did most of the subsequent revisions. I regularly discussed this work with my supervisor Prof. Volker Tresp, who supervised the finds of this work. All co-authors discussed the results periodically and assisted in polishing the final manuscript.

Chapter 4 presents this work.

- Yunpu Ma, Volker Tresp. A Quantum Machine Learning Algorithm for Knowledge Graphs. under review. arXiv preprint arXiv:2001.01077, without doi.

I conceived of the presented idea, performed all implementations, and developed all theories. I wrote the initial manuscript and did most of the subsequent revisions. I regularly discussed this work with my supervisor Prof. Volker Tresp, who supervised the finds of this work. All co-authors discussed the results periodically and assisted in polishing the final manuscript.

Chapter 5 presents this work.

- Yunpu Ma, Yuyi Wang, and Volker Tresp. Causal Inference under Networked Interference and Intervention Policy Enhancement. under review. arXiv preprint arXiv:2002.08506, without doi.

I conceived of the presented idea, performed all implementations, and developed all theories. I wrote the initial manuscript and did most of the subsequent revisions. Yuyi Wang pointed me to the theory of concentration inequality for networked variables. I regularly discussed this work with my supervisor Prof. Volker Tresp, who supervised the finds of this work. All co-authors discussed the results periodically and assisted in polishing the final manuscript.

Chapter 6 presents this work.

Chapter 1

Introduction

1.1 Motivation

Knowledge representation is an essential subdiscipline of artificial intelligence (AI) [98, 62] with focuses on translating external information and human knowledge of the world into machine-understandable language, such that an artificial agent can utilize this knowledge and undertake specific complex tasks. Different schemes for relational knowledge representation have been proposed in the history of artificial intelligence, such as the Fame [77] that stores knowledge in substructures and a semantic network [104] that adopts a conceptual dependency graph to describe semantically structured knowledge. Relational knowledge representation can also be augmented with logic rules, hence bringing explainable and reliable knowledge into current artificial intelligence approaches.

Current attempts for artificial intelligence are based on neural networks and designed to extract meaningful latent representations from raw input features for subsequent tasks. Depending on the tasks, specific neural network architectures have been proposed, e.g., Feedforward Neural Network [100] for supervised learning, Convolutional Neural Network (CNN) [63] for image classification, Recurrent Neural Network (RNN) [25, 48] for sequence modeling tasks, and Graph Neural Network (GNN) [46] for modeling graph-structured raw data. Neural networks are flexible deep learning frameworks for solving various tasks, but they cannot explicitly leverage expert knowledge, and they always require massive human-labeled data. Therefore, explicit and well-structured knowledge representations are supposed to be perfect complements to machine learning and deep learning.

The semantic Web [11] is an example of explicit and relational knowledge representation, which creates a standard framework for interlinking Internet data, such that these

machine-readable data can be shared and extended across different domains and systems. To develop the Semantic Web for interchangeable data, technology such as Resource Description Framework (RDF) [58] has been proposed, where links between two resources are expressed in the subject-predicate-object format, also known as the SPO triples. Semantic Web is an ambitious project proposed by Tim Berners-Lee, inventor of the World Wide Web, and remains to be fully accomplished. On the other hand, the Knowledge Graph (KG), as a variant of Semantic Web, inherits the RDF format and describes existing facts as relationships between entities using the subject-predicate-object triples. Due to the explicit semantic meaning of the subject-predicate-object triples, facts stored in a knowledge graph are also referred to as semantic triples.

Recently, various relational knowledge bases, or knowledge graphs, have been created by different companies, research institutes, and communities for their purposes of use, especially for facilitating AI systems. For example, Google Knowledge Graph [103] gathers billions of facts from a variety of resources to improve search engine's quality and to assist the question-answering service. YAGO [105] is a huge semantic knowledge base about people, countries, and organizations, and helps to build the Watson cognitive platform [28]. Moreover, Freebase [12] is a community-oriented knowledge graph that is extracted from Wikipedia and augmented by its community members. A social network can be regarded as a domain-specific knowledge graph with nodes representing individuals and edges the friendship relations among individuals. Hence, based on the social graph nature of Facebook, a social graph search engine was developed by Facebook to enhance the friend recommendation service.

Knowledge graphs provide a structured and declarative representation of knowledge and support the intelligent reasoning on knowledge through learning. The purpose of learning on knowledge graphs is to obtain features or representations of entities and encode complex relational structures into these features as a form of knowledge representation. After learning on knowledge graphs, statistical inference can be conducted using learned representations for the tasks such as missing link prediction, entities' attributes prediction, and entities classification [83]. This kind of learning and reasoning belongs to the field of statistical relational learning (SRL) [59]. Numerous statistical learning algorithms have been developed for modeling knowledge graphs such as RESCAL [85] and DistMult [125]. In comparison with the rule-based reasoning system, scalability and generalizability are the main advantages of learning-based algorithms. It is feasible to generalize statistical learning algorithms to knowledge graphs with millions of entities. Therefore, in this dissertation,

we mainly develop learning-based algorithms to solve reasoning and prediction tasks.

So far, knowledge graphs are introduced as relational databases that store relationships between entities as semantic triples. Hence, these relational databases are referred to as semantic knowledge graphs. In most circumstances, however, the world is dynamic, and facts of the world might change over time. For instance, one patient is recovered from a disease, or the political relationship between two countries becomes tense. In order to declaratively describe the changing facts, episodic knowledge graphs were introduced, which are also known as temporal or time-dependent knowledge graphs. They are large-scale event databases used to represent time-evolving and multi-relational data. Examples of standard episodic knowledge graphs are GDELT [64] and ICEWS [13], which were created to keep global events such as interactions between countries and organizations. Moreover, time-dependent facts are stored in the episodic knowledge graphs as quadruples with additional timestamps. One other subject of this dissertation is to develop statistical learning algorithms on episodic knowledge graphs and to perform probabilistic inference on them.

Semantic and episodic knowledge graphs store information in a declarative form. They are believed to be the most abstract and succinct way to represent facts and human knowledge. Therefore, an analogy between knowledge graphs and human’s cognitive memory functions was first observed and proposed in [111, 112]. In particular, the papers suggest that semantic knowledge graphs are technical realizations of semantic memory, and episodic knowledge graphs correspond to episodic memory. Semantic and episodic memories are long-term declarative memories that support each other. Hence, it is expected that their technical realizations, the semantic and episodic knowledge graphs, are interdependent. [113, 69] demonstrate the above idea and show that a semantic knowledge can be derived from an episodic knowledge via marginalization. The ideas regarding knowledge graphs from cognitive perspectives are pushed forward recently in the effort [114], where knowledge graphs are suggested to support the semantic decoding of the perception as well as memory consolidations. Also, in this dissertation, we improve a previous cognitive architecture for the associative memory, the holographic reduced representation [89], and apply the novel framework to large-scale knowledge graphs.

Information is extracted from various text resources and aggregated into a knowledge graph such as from webpages, newspaper articles, and scientific reports, resulting in consistently growing knowledge graphs with an increasing number of semantic triples and unique entities. The growing number of semantic triples and entities slows down the training and

inference on knowledge graphs. In the second part of this dissertation, we, therefore, investigate quantum algorithms that can accelerate the learning procedure and the reasoning on knowledge graphs, which can be regarded as our second significant contribution. Primarily, we propose two types of quantum algorithms for reasoning on knowledge graphs. The first approach is a training-based algorithm using a hybrid quantum-classical building block, called the variational quantum circuit [78]. This hybrid approach adopts a circuit-centric [101] parametrized quantum circuit to estimate the score functions of semantic triples, such that the computational complexity for evaluating the score functions can be dramatically reduced. Besides, it employs a classical unit for storing and updating the parameters in the variational circuit. Furthermore, we show that there exists a heuristic quantum subroutine that can quadratically accelerate the reasoning.

The second approach is a sampling-based quantum algorithm, which can be regarded as the quantum counterpart of classical tensor singular tensor decomposition [19] (tensor SVD). The quantum tensor SVD approach employs a quantum memory architecture, the Quantum Random Access Memory [33], and quantum subroutines such as density matrix exponentiation [67], quantum phase estimation [57], and quantum singular value projection [54], realizing an exponential acceleration for the reasoning on knowledge graphs. As one theoretical contribution, we provide rigorous conditions, under which tensor singular value decomposition is plausible for the tensor completion task, and under which the quantum counterpart is feasible.

In the last part of this dissertation, we investigate causal inference in complex relational domains. Major subjects studied in the causal inference are, for example, identifying the causal direction from observed data and predict the potential outcomes from observational or experimental studies. We will focus on the framework for inferring potential outcomes, also known as the Neyman-Rubin causal model [95, 97] and generalize it to the relational domain. In the relation domain, the outcome of an individual is not only affected by the treatment assignment to this individual but also by the treatment assignments or responses of its neighboring individuals. This causal interference phenomenon is referred to as the spillover effect in economics or peer effect in social science [4]. For instance, a student's academic performance depends not only on whether it attends a tutorial class, which can be regarded as a treatment being assigned to the student, but also on whether its friends attend a tutorial class and its friends' performances.

Such a problem, inferring causal effects under network interference, is a challenging yet intriguing research topic and has attracted attention from the causal inference community.

To capture the neighboring influence, we adopt graph neural networks (GNNs) as powerful aggregators for aggregating information of neighboring feature vectors and develop causal estimators upon them. We provide a heuristic error bound for these novel GNN-based causal estimators and show their superior performance. After obtaining the optimal causal estimators, we further learn a policy network to maximize the average welfare on the network by reassigning intervention decisions. As a novel theoretical contribution, we provide policy regret for policy networks that employ GNN-based causal estimators with and without treatment capacity constraint.

This cumulative dissertation is organized as the following. Chapter 1 serves as a general introduction of this dissertation and provides necessary backgrounds. Section 1.1 motivates the subjects investigated and provides an overview. Section 1.2 reviews different machine learning approaches for modeling semantic knowledge graphs, while Section 1.3 introduces episodic knowledge graphs and reviews tensor decomposition and point process methods for modeling episodic knowledge graphs. In Section 1.4, we discuss the cognitive perspectives of semantic and episodic knowledge graphs and their connections to declarative memories. Section 1.5 further introduces a cognitive architecture for the associative memory, the holographic reduced representation, and discusses its improvement and application on semantic knowledge graphs. Section 1.6 and 1.7 provide necessary backgrounds for understanding the two quantum algorithms for reasoning on semantic knowledge graphs. Moreover, Section 1.8 introduces the task of causal inference under interference and proposes the GNN-based causal estimators. Chapters 2, 3, 4, 5, and 6 present our published works and works under review. Chapter 7 gives a summary and points out further research directions as continuations of presented works.

1.2 Modeling of Semantic Knowledge Graphs

1.2.1 Introduction of Semantic Knowledge Graphs

We first provide a brief introduction of semantic knowledge graphs and representation learning of semantic knowledge graphs. A set of semantic facts that describe the relations between entities builds a semantic knowledge graph. A semantic fact in the Resource Description Framework (RDF) [61] is represented as a triple (*subject*, *predicate*, *object*), which is also referred to as a semantic triple. Figure 1.1 (a) illustrates a small fragment of a semantic knowledge graph. It keeps, for instance, the location of the city of Hamburg

and describes this fact as the semantic triple $(Hamburg, CityOf, Germany)$. Freebase [12], Wikidata [120], and YAGO [73] are standard knowledge graphs that are widely used in AI systems. Similar to the preference prediction and item recommendation using the preference matrix, by modeling semantic knowledge graphs, we can infer implicit and missing knowledge from observed semantic triples [83]. Since knowledge graphs contain noisy facts and missing links after automatic extraction, the inference task becomes a standard method for testing the modeling algorithms developed on them.

Besides the database format for storing semantic triples, a knowledge graph can also be viewed as a directed graph with labeled edges representing predicates and nodes being the entities. Note that a directed and labeled graph is equivalent to a three-dimensional tensor with one dimension for subjects, one for objects, and one for predicates. Hence, another view of a knowledge graph is a three-dimensional sparse tensor, with entries indicating the truth values of corresponding semantic triples. A tensor decomposition method for modeling the three-dimensional semantic tensor has been proposed in [85]. This tensor decomposition method, also known as RESCAL, is demonstrated in Figure 1.1 (b). RESCAL decomposes the three-dimensional semantic tensor into lower-dimensional vector representations of entities and matrix representations of predicates, which can be used in subsequent tasks. Other tensor decomposition algorithms that have been applied to knowledge graphs are, for example, PARAFAC [44], HOSVD [22], and the Tucker model [117].

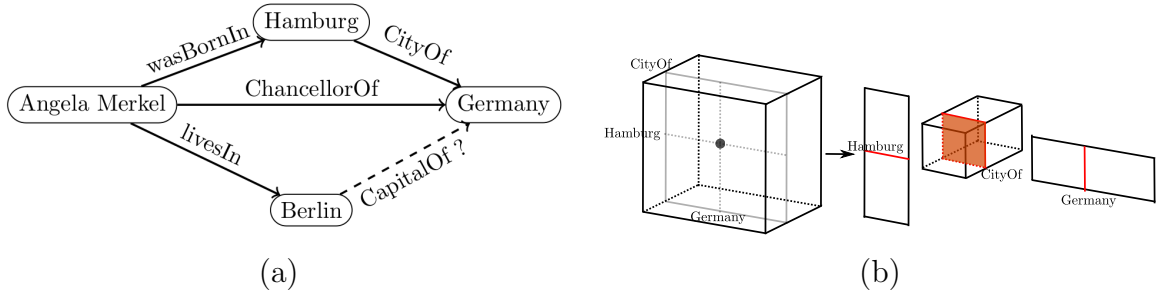


Figure 1.1: (a) A fragment of a semantic knowledge graph with semantic triples, e.g., $(Hamburg, CityOf, Germany)$ and $(Berlin, CapitalOf, Germany)$, etc. (b) illustrates the tensor view of a semantic knowledge graph and the RESCAL [85] model for semantic tensor decomposition. In RESCAL, each entity has a vector representation, and each predicate possesses a matrix representation. The score function of each semantic triple is then obtained via the vector-matrix-vector multiplication.

1.2.2 Relational Learning for Semantic Knowledge Graphs

We provide a brief introduction to different modeling methods of semantic knowledge graphs. We let \mathcal{E} denote the set of entities with the size N_e , and \mathcal{P} the collection of predicates with the size N_p . We write the three-way semantic tensor as χ with entries x_{spo} indicating the truth value of the corresponding semantic triple (s, p, o) , i.e., $\chi \in \{0, 1\}^{N_e \times N_p \times N_e}$. As mentioned previously, the main idea of representation learning is to obtain low-dimensional representations of entities and predicates that can well capture the global patterns in the knowledge graph. Hence, we assume unique latent representations for each entity and predicate. We let \mathbf{a}_{e_i} , $i = 1, \dots, N_e$, denote entity representations, and \mathbf{a}_{p_i} , $i = 1, \dots, N_p$, predicate representations. Note that given the triple (s, p, o) , we also interchangeably use \mathbf{a}_s , \mathbf{a}_p , and \mathbf{a}_o as representations of the subject, predicate, and object, respectively. We further introduce a companion tensor H sharing the same shape as χ with entries η_{spo} . Statistical modeling of the semantic tensor χ assumes that the probability of an entry being true is given by $\Pr(x_{spo} | \eta_{spo}) = \sigma(\eta_{spo})$, where the sigmoid function is defined as $\sigma(x) := \frac{1}{1+e^{-x}}$.

The tensor element η_{spo} assigns a score to the triple (s, p, o) . Hence it also referred to as the score function of the triple (s, p, o) . The score function is a function of latent representations, and usually, each model is associated with a unique way of composing the score function. For instance, the Tucker tensor decomposition with rank R is defined as

$$\eta_{spo} = \sum_{r_1, r_2, r_3=1}^R \mathcal{G}_{r_1, r_2, r_3} \mathbf{a}_{s, r_1} \mathbf{a}_{p, r_2} \mathbf{a}_{o, r_3},$$

where we assume that each entity and predicate has an R -dimensional vector representation; $\mathcal{G} \in \mathbb{R}^{R \times R \times R}$ represents the core tensor. Moreover, the RESCAL model assumes vector representations of entities and matrix representations of predicates (see Figure 1.1 (b)) whose score function is composed as

$$\eta_{spo} = \sum_{r_1, r_2=1}^R \mathbf{a}_{s, r_1} \mathbf{a}_{p, r_1, r_2} \mathbf{a}_{o, r_2},$$

with $\mathbf{a}_{e_i} \in \mathbb{R}^R$, for $i = 1, \dots, N_e$, and $\mathbf{a}_{p_i} \in \mathbb{R}^{R \times R}$, for $i = 1, \dots, N_p$.

In addition to tensor decomposition approaches, several compositional and translational methods for the score function have been proposed in [125, 116, 84]. One simple compositional model DistMult [125] uses a tri-linear dot function and defines the score function as

$$\eta_{spo} = \sum_{r=1}^R \mathbf{a}_{s, r} \mathbf{a}_{p, r} \mathbf{a}_{o, r},$$

where it assumes that all entities and predicates are R -dimensional real-valued vectors. Moreover, ComplEx [116] is proposed as an extension of DistMult by using complex-valued vector representations. The score function for ComplEx is defined as

$$\eta_{spo} = \Re\left(\sum_{r=1}^R \mathbf{a}_{s,r}, \mathbf{a}_{p,r}, \bar{\mathbf{a}}_{o,r}\right),$$

where the bar indicates the complex conjugate operation, and \Re represents the real part of a complex number.

The semantic tensor is usually an extremely sparse tensor whose nonzero entries take value one and represent real or observed facts. Under the *closed world assumption* [83], unobserved semantic triples are assumed to be false and registered as zero entries into the semantic tensor. Latent representations of entities and predicates are obtained by minimizing a loss function and using a dataset that consists of both true and false semantic triples. Consider a training dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^m$ with m samples, where x_i represent semantic triples and y_i the corresponding truth values. The loss function can be chosen as, for instance, a binary cross-entropy loss that is defined as

$$\mathcal{L} = -\frac{1}{m} \sum_{i=1}^m (y_i \log(\sigma(\eta_{x_i})) + (1 - y_i) \log(1 - \sigma(\eta_{x_i}))),$$

where σ is the sigmoid function, and η_{x_i} represents the score function of the semantic triple x_i .

Inferring implicit knowledge, or missing link prediction, is only one of the applications with knowledge graphs. Google uses knowledge graphs to enhance the quality of its search engine. Semantic knowledge graphs can also provide rich and structured knowledge to pre-trained neural language models to improve natural language understanding [127]. Complicated conjunctive logical queries can also be conducted on knowledge graphs using learned representations, e.g., an existential query “What drugs can target proteins that are associated with both mutations A and B?” on a biological knowledge base [40]. Besides, knowledge graphs can facilitate fact-checking using path-based reasoning and extracted arguments by two debating agents [47]. In summary, modeling knowledge graphs and using them for the subsequent tasks are exciting and imaginative research topics, and there are more can be explored.

1.3 Modeling of Episodic Knowledge Graphs

1.3.1 Relational Learning for Episodic Knowledge Graphs

Episodic knowledge graphs, also known as temporal or time-dependent knowledge graphs, are large-scale event databases, which can describe temporally evolving multi-relational data. An episodic knowledge graph can be viewed as a sequence of semantic knowledge graphs accompanied by timestamps. Hence, the entries of an episodic knowledge graph are quadruples composed of subject, predicate, object, and timestamp, also denoted as (s, p, o, t) for short. For instance, Global Database of Events, Language, and Tone (GDELT) [64] and Integrated Crisis Early Warning System (ICEWS) [13] are two available event-based temporal knowledge graphs that have been drawing attention in the community. As the name suggests, the data repository GDELT registers evolving knowledge about interactions between countries and organizations across the globe. The ICEWS data repository contains information about national and international crises.

In the last section, we have reviewed several representation learning models of semantic knowledge graphs, including tensor decomposition and compositional models. For instance, representative tensor decomposition models are Tucker and the RESCAL model, and DistMult and ComplEx are representative compositional models. In the pioneering work [69], we are the first to investigate representation learning models for episodic knowledge graphs. To generalize the semantic models to episodic knowledge graphs, we introduce unique latent representations for timestamps. For instance, Figure 1.2 (a) shows the extension of three-way Tucker to four-way Tucker, where a four-dimensional core tensor replaces the three-dimensional core tensor with one dimension representing the timestamp. Figure 1.2 also shows the extensions of RESCAL and ComplEx and their applications to episodic knowledge graphs. In particular, the ConT model, which is the extension of RESCAL, shows a superior memory capacity since each timestamp is associated with a three-dimensional core tensor in this tensor decomposition model. The high dimensionality of the timestamp representations not only results in superior memory capacity of the model, but it also facilitates the mapping from episodic knowledge to semantic knowledge.

A similar approach has been proposed in [21], which develops an extension of the translational model of static knowledge graphs by introducing latent representations for timestamps. The main disadvantage of these approaches is a restricted generalization ability to unobserved timestamps. In other words, these models are limited to the completion tasks on episodic knowledge graphs with observed timestamps in the training dataset, and they

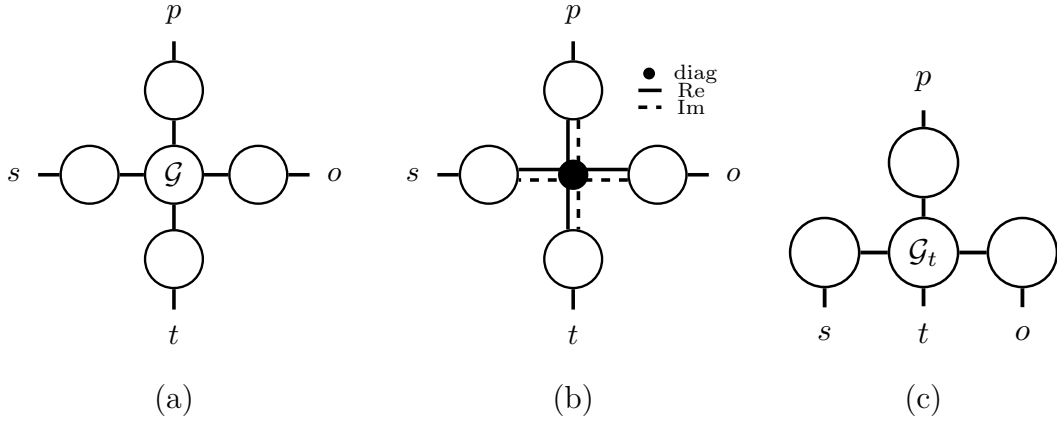


Figure 1.2: Illustrations of (a) Tucker, (b) ComplEx, and (c) ConT models for representation learning of episodic knowledge graphs introduced in [69]. The Tucker model realizes a four-way tensor decomposition with dimensions representing subjects, predicates, objects, and timestamps. The four-way ComplEx model is an extension of the ComplEx model for semantic knowledge graph. ConT can be regarded as a generalization of the RESCAL model, where \mathcal{G}_t indicates the three-dimensional tensor representation of a timestamp.

cannot predict future events. To resolve these issues, [31] introduces latent representations for each character of the timestamp, such that latent representations can be generalized to unobserved timestamps. Know-Evolve [115], on the other hand, provides a novel deep learning architecture that allows reasoning over time and future events prediction. Know-Evolve models the non-linearly evolving events as a nonparametric point process whose intensity function is characterized by time-dependent entity representations.

1.3.2 Graph Hawkes Network for Modeling Episodic Knowledge Graphs

It is advantageous to introduce the nonparametric point process for modeling temporal knowledge graphs. For instance, modeling events as point process can return the intensity or the probability of the occurrence of an event over continuous time, which makes it different from the aforementioned temporal knowledge graph models using discrete timestamps. In particular, the intensity function can be interpolated between timestamps and extrapolated to the future, such that the occurrence time of future events can be predicted. However, as pointed out by the follow-up work [42], Know-Evolve cannot deal with concurrent entities and event interactions within the same timestamp.

To resolve the problem of concurrent events, we proposed a Graph Hawkes Network (GHN) for future events prediction in [42]¹. The GHN model captures the underlying interactions between entities and events using a Graph Neural Network and estimates the intensity function of an event from the historical events via a multivariate Hawkes process. To be more specific, the GNN module aggregates information from concurrent events, and the intensity functions in the Hawkes process is returned by a continuous-time LSTM [76] that uses the aggregated information from the GNN module as inputs. Note that each event type in a temporal knowledge graph corresponds to a directed and labeled edge, namely unique event types are unique semantic triples by ignoring the time information of quadruples. From another perspective, a temporal knowledge graph can be considered as an event stream with many event types. Hence, another superior advantage of the GHN model is its flexibility of modeling event stream with a large number of event types.

Hawkes Process [45] is defined to be a self-exciting multivariate point process for modeling sequential discrete and inter-dependent events that occur over continuous time. It is a self-exciting process in which the occurrence of an event can raise the conditional intensity of other events. The conditional intensity function of one event type at instance t depends explicitly on previously happened events, and it takes the form

$$\lambda_k(t) = \mu_k + \sum_{h:t_h < t} \alpha_{k_h, k} \exp(-\delta_{k_h, k}(t - t_h)),$$

where μ_k is the background intensity of event type k , $\alpha_{j, k}$ characterizes the degree of excitation of event type j on type k , and $\delta_{j, k}$ describes the exponential decay rate of the excitation with time. The conditional probability density function $p_k(t)$, which describes the probability that the next event with type k will occur during the interval $[t, t + dt)$ conditioned on the past events, can be calculated using survival analysis theory [1]. The density function is defined to be the product of the intensity function and the probability that no event of any type happens after the latest event, namely

$$p_k(t) = \lambda_k(t) \exp\left(-\int_{t_L}^t \sum_k \lambda_k(s) ds\right),$$

where t_L represents the time of the latest event that happened before t .

In the context of modeling episodic knowledge graphs via Hawkes process, we let $e_i = (e_{s_i}, e_{p_i}, e_{o_i}, t_i)$ denote an event happened at time t_i having type $(e_{s_i}, e_{p_i}, e_{o_i})$, where e_{s_i} , e_{p_i} , and e_{o_i} are the subject, predicate, and object of the event e_i , respectively. Without going

¹as a second author work with significant contribution

into details of the GHN model, we discuss the reasoning ability of it. A plausible task on the temporal knowledge graph is to infer the missing entities of events at a future timestamp t_i conditioned on observed histories, e.g., to predict the missing subject $(?, e_{p_i}, e_{o_i}, t_i)$ or the missing object $(e_{s_i}, e_{p_i}, ?, t_i)$. Besides entities predictions, the GHN model can infer when an event type will happen, namely $(e_{s_i}, e_{p_i}, e_{o_i}, ?)$, which is also known as the time prediction task.

More concretely, given query $(e_{s_i}, e_{p_i}, ?, t_i)$, we use the GHN model to estimate the intensity functions $\lambda(e_o|e_{s_i}, e_{p_i}, t_i, \mathcal{H})$ of all candidate objects e_o and locate the possible ones, where \mathcal{H} represents observed histories, and the intensity function is conditioned on the query and the history \mathcal{H} . Also, for the time prediction task $(e_{s_i}, e_{p_i}, e_{o_i}, ?)$, we can obtain a similar conditional intensity function $\lambda(t|e_{s_i}, e_{p_i}, e_{o_i}, \mathcal{H})$ from GHN, which describes the intensity of event type $(e_{s_i}, e_{p_i}, e_{o_i})$ at instance t given the history \mathcal{H} . The corresponding density function of this event type reads

$$p(t|e_{s_i}, e_{p_i}, e_{o_i}, \mathcal{H}) = \lambda(t|e_{s_i}, e_{p_i}, e_{o_i}, \mathcal{H}) \exp\left(-\int_{t_L}^t \lambda(\tau|e_{s_i}, e_{p_i}, e_{o_i}, \mathcal{H}) d\tau\right).$$

The expected occurrence time of the given event type can then be estimated as

$$\int_{t_L}^{\infty} t p(t|e_{s_i}, e_{p_i}, e_{o_i}, \mathcal{H}) dt.$$

More details on the model architecture and experimental results are provided in [42].

1.4 Cognitive Perspectives of Knowledge Graphs

1.4.1 Knowledge Graphs and Cognitive Memories

Declarative and nondeclarative memories are two components of the brain's long-term memory, where declarative memory can be further divided into semantic memory, episodic memory, and autobiographic memory [32]. Declarative memory refers to the memory of facts and events which can be recalled and described with language. Whereas semantic memory refers to conscious recollection of factual knowledge and concepts, episodic memory is an intentional retrieval of previous events with their spatial and temporal contexts. The difference between episodic memory and autobiographic memory is that autobiographical memory is only associated with specific personal experiences. Nondeclarative memory, on the other hand, is an unconscious memory, including perceptual and procedural memories, which are related to the acquisition of better skills and formation of habits. Figure 1.3 demonstrates a simple classification of the brain's different types of memory.

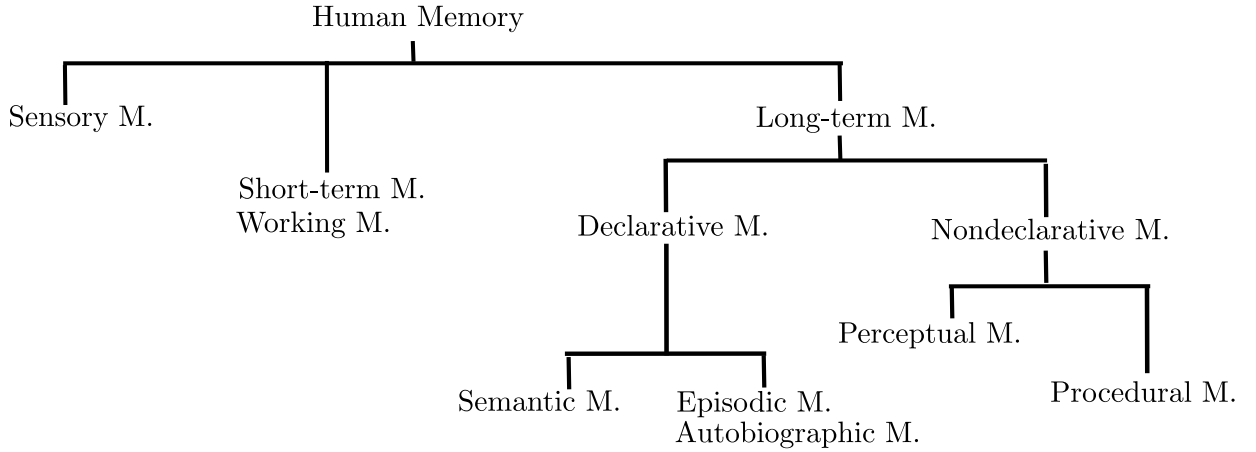


Figure 1.3: Basic classification of human memory functions.

A technical realization of semantic memory is a collection of semantic triples, e.g., the triple *(California, locatedIn, USA)* storing the factual knowledge that California locates in the USA. The collection of quadruples, or semantic triples accompanied with timestamps, realizes the episodic memory, e.g., the quadruple *(Jack, diagnosedWith, Diabetes, Feb10)* recording the fact that Jack was diagnosed with diabetes on February 10. A semantic knowledge describing the current health condition of Jack can be derived from the quadruple by ignoring the temporal information, namely *(Jack, diagnosedWith, Diabetes)*.

As motivated previously, a 3-way semantic tensor, with dimensions for subjects, predicates, and objects, is a suitable representation of semantic memory, while a 4-way episodic tensor with additional timestamp dimension is suitable a data representation of episodic memory. The tensor view of declarative memory is not efficient and compressed. Hence, a biologically more plausible view of declarative memory is proposed in [113]. In this framework, declarative memory tensors are first decomposed, and each generalized entity, predicate, and timestamp obtains a unique latent representation, such that memory tensors can be approximately reconstructed from latent representations. The tensor decomposition approach provides a compressed form of declarative memory, and more advanced, knowledge generalization becomes plausible by inferring new semantic triples or episodic quadruples using latent representations.

The tensor decomposition framework for declarative memories is biologically plausible since generalized entities and distributed representations can find their counterparts in the brain, which are widely studied in cognitive neuroscience. Entities for abstract and symbolic concepts are encoded as concept cells resided in the medial temporal lobe (MTL)

- the hippocampus and its surrounding cortex [91]. Concept cell represents one neuron or a separate assembly of neurons that become activated when perceiving a specific concept. The activation of the concept cell corresponding to a specific concept can be triggered by different aspects of the same concept, e.g., the visual or acoustic features of the concept, since there exist links between the concept cell and cortical areas that store different aspects of the corresponding concept. For example, concept cell is also called as Jennifer Aniston cell, since it might be invoked by Jennifer’s appearance, voice, or even the movies she starred.

Furthermore, the entity representations in the tensor decomposition framework resemble the distributed representations of concepts stored in different cortical areas. Reversely, the activation of a concept cell brings the conscious retrieval of various attributes of the corresponding concept. The localized storage of concepts and distributed storage of conceptual representations form a flexible system of long-term memory, which even contributes to perception, language, and thought [55].

Cognitive studies suggest that semantic and episodic memories are interdependent both at encoding and retrieval phases [36]. Baddeley [8] argues that semantic memory might arise from blurred episodic memory by losing temporal information, in the sense that repeatedly experienced episodic events become consolidated, and during conscious retrieval, only decontextualized events can be recalled. For example, by consistently noticing that Jack is diagnosed with diabetes on February 10, one becomes aware that Jack has diabetes. As proposed in [113, 112], in the tensor decomposition framework of declarative memories, the technical realization of decontextualization of episodic memory and the transition from episodic memory to semantic memory are implemented via marginalization in the time dimension. The marginalization is performed using the latent representations of timestamps. However, since semantic memory reflects the factual knowledge of *current* timestamp, marginalization should be implemented while minding the end timestamps of repeatedly experienced episodic events [69].

Until now, we have assumed that the distributed representations of timestamps only come from the tensor decomposition of the episodic tensor, and the semantic decoding of episodic events is realized by marginalization over time dimension. However, according to the definition of episodic memory, it refers to the memory of specific events and related contexts, such as spatial, temporal, visual information, and associated emotion, etc. Previous theory of episodic memory, such as the Hippocampal Memory Indexing Theory (HMIT) [109], suggests that episodic events are stored by forming time indices in the brain

that connect to the representation layer. Activation of the time indices brings the recollection of previous experiences. However, the hippocampal memory indexing theory is subsymbolic, which contradicts the fact that episodic memory is declarative.

1.4.2 Cognitive Architecture for Semantic Decoding

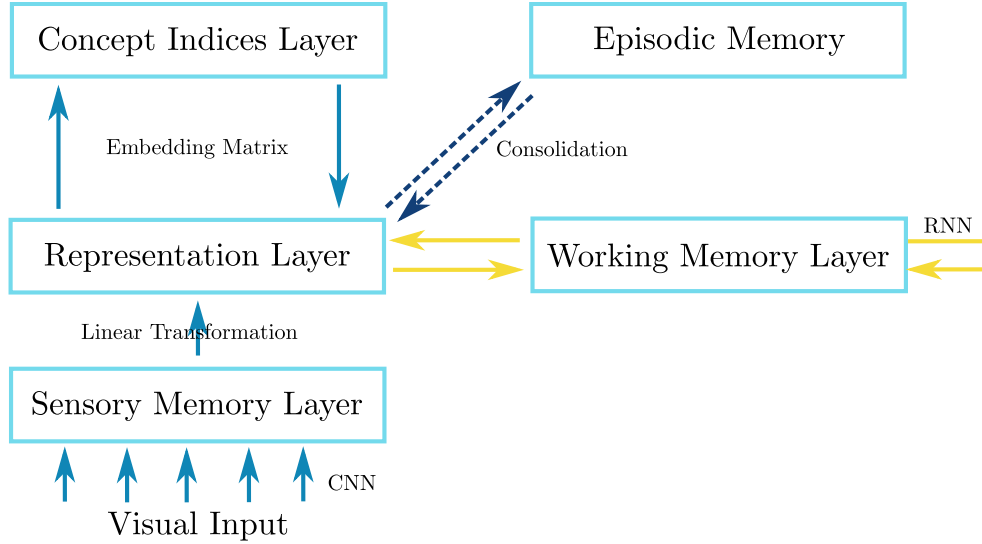


Figure 1.4: A cognitive neural architecture that describes the sequential semantic decoding for perception, which is proposed in [114].

To extract the declarative information from episodic events, a cognitive neural architecture for sequential semantic decoding from perception is proposed in [114]. Figure 1.4 sketches the basic idea of this cognitive architecture for scene understanding. Visual inputs are first processed and passed through a convolutional neural network and stored in the sensory memory layer as visual representations. Representations for perceived entities are initially obtained from a linear transformation of the memory stored in the sensory memory layer and then reactivated via the connections between concept neurons and the representation layer. Visual relations are recognized by iteratively sampling the semantic triples with the help of a recurrent neural work, which mostly resembles the working memory. Furthermore, the scene can be consolidated and stored as an episodic memory. More details of this cognitive neural architecture for modeling the human memory system can be found in [114].

Semantic decoding utilizes semantic memory and facilitates the interpretation of perception, and it is believed to be a unique capacity of the human. Through semantic

decoding, semantic triple statements of the visual perception are generated, which can compose the thought for verbally describing the visual perception. Notice that during the semantic decoding, the semantic memory serves as prior knowledge for creating triple statements and facilitates the efficient formation of episodic memory [9, 10]. Therefore, the conscious and declarative recollection of a piece of episodic memory might be realized by activating the time index, which triggers the activation of the corresponding representation layer, and semantic decoding then follows up. Moreover, repeated recollections or experiences of the same episodic event lead to memory consolidation and, eventually, a transition from episodic memory to semantic memory. A summary of the cognitive perspectives of knowledge graphs is presented in Figure 1.5.

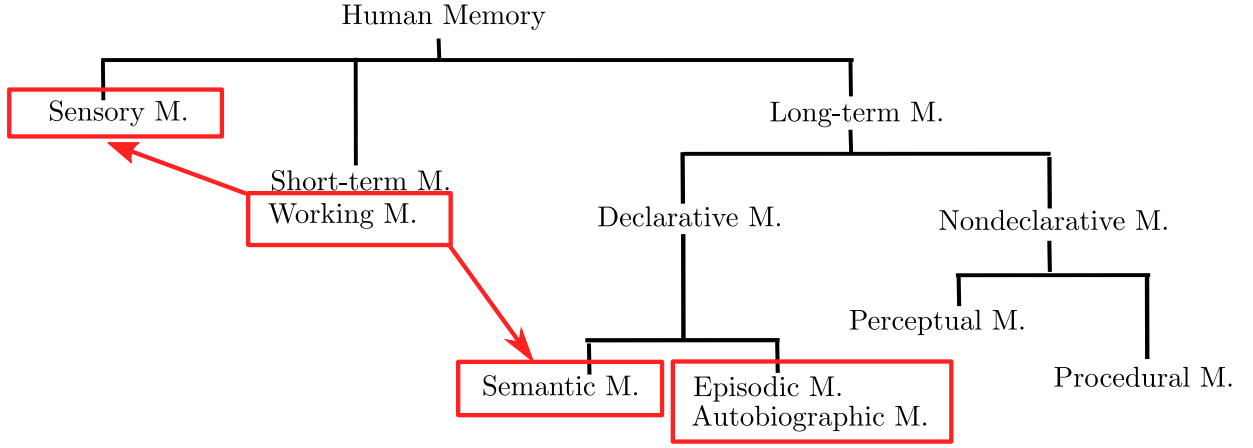


Figure 1.5: An overview of our cognitive perspectives of knowledge graphs. The semantic knowledge graph achieves a technical realization of semantic memory, while episodic memory is a technical realization of episodic memory. A semantic decoder that incorporates long-term semantic memory, sensory memory, and short-term working memory provides a declarative property of episodic memory.

1.5 Holographic Reduced Representation and Holistic Representation

1.5.1 Holographic Reduced Representation

The tensor decomposition framework for modeling declarative memories is a learning-based cognitive architecture where latent representations of generalized entities are obtained via

tensor decomposition. It also serves as a vector symbolic architecture (VSA) for structured associative memory in the sense that an incomplete entry can be approximately completed using latent representations. For example, the associative query $(s, p, ?)$ is answered by ranking the scores objects, realizing an architecture for associative recollection. Associative memory is an essential component of artificial intelligence since connections between seemingly unrelated concepts can be stored in the associative memory, e.g., paired stimuli and responses. Various approaches for modeling the associative memory have been proposed, starting from the Hopfield's network [49], which is a learning-based architecture. Among various vector symbolic architectures for simulating associative memory, holographic associative memory (HAM) was first studied in [30, 121], which suggests that the holographic storage of many stimulus-response pairs might be related to the working principles of the human brain.

Holographic associative memory was further modified to the Holographic Reduced Representation (HRR) proposed in [89]. In the HRR cognitive structure, each item or symbol is associated with a vector representation. An association between two items is reduced to a single vector of the same dimension via a vector binding operation. To ensure the reduction is reversible, circular convolution and correlation are chosen as the binding operations. The reduced representation for the item association forms a memory trace, and multiple memory traces, i.e., multiple associations, can be compressed into a single memory trace via superposition, which is, defined mathematically, an addition in the same vector space. Two associated items are called a cue-filler pair. Through the reversible vector binding operation, encoded cue-filler pairs can be approximately retrieved from this single memory trace. However, the restored representation is usually distorted due to the interference caused by the superposition of multiple associations. Therefore, to identify the retrieved item, a clean-up post-process is required in the HRR cognitive structure.

Usually, multiple associations are stored holographically in a memory trace, e.g., a memory trace of a horse might encode the breed, color, and height of the horse. Intuitively, the more associative pairs encoded in a memory trace, the less accurate associations can be retrieved. The maximal number of cue-filler pairs encoded in a single memory trace, which can still be approximately retrieved, is referred to as the memory capacity of the cognitive structure for associative memory. Before understanding how the circular convolution- and correlation-based binding operation affects the memory capacity, especially when it is applied to directed knowledge graphs, we first discuss another critical factor, the initialization of vectors in the VSA. Recall that, in holographic reduced representation, distributed rep-

representations of items are initialized by elementwise sampling from a Gaussian distribution. It has been shown in [89] that the memory capacity of the HRR architecture depends on the degree of pairwise quasi-orthogonality of initialized random vectors.

1.5.2 Quasi-Orthogonality

The concept of near orthogonality was studied in [24], which informally states that most vectors in an ensemble of independently sampled random vectors are nearly orthogonal. A more rigorous mathematical definition of near orthogonality, or quasi-orthogonality, has only recently been addressed in [16, 17], where, given an ensemble of Gaussian random vectors, the asymptotic distribution function of the cosine of pairwise angles is derived. The more mathematical term, ϵ -orthogonality, is formally defined as follows.

Definition 1. *A set of n vectors $\mathbf{X}_1, \dots, \mathbf{X}_n$ is said to be pairwise ϵ -orthogonal, if $|\langle \mathbf{X}_i, \mathbf{X}_j \rangle| < \epsilon$ for $i, j = 1, \dots, n$, $i \neq j$, where $\epsilon > 0$, and $\langle \cdot, \cdot \rangle$ denotes the inner product in the vector space.*

We assume that $\mathbf{X}_1, \dots, \mathbf{X}_n$ are q -dimensional vectors with elements randomly sampled from the normal distribution $\mathcal{N}(0, 1)$. Let Θ_{ij} denote the angle between vectors \mathbf{X}_i and \mathbf{X}_j and let $\rho_{ij} := \cos \Theta_{ij} \in [-1, 1]$. The distribution function of random variables ρ_{ij} in the large n limit, $n \rightarrow \infty$, is derived in [16, 81] and revisited in the following Lemma.

Lemma 1. *Consider ρ_{ij} as defined above. Then $\{\rho_{ij} | 1 \leq i < j \leq n\}$ are pairwise i.i.d. random variables with the following asymptotic probability density function*

$$g(\rho_G) = \frac{1}{\sqrt{\pi}} \frac{\Gamma(\frac{q}{2})}{\Gamma(\frac{q-1}{2})} (1 - \rho_G^2)^{\frac{q-3}{2}}, \quad |\rho_G| < 1, \quad (1.1)$$

with fixed dimensionality q , where the subindex in ρ_G indicates that random vectors are sampled from Gaussian distribution.

Since the concept of holographic reduced representation, it had mainly been tested on small toy datasets for associative memory tasks. The main difficulties of applying the HRR architecture to large-scale knowledge graphs are a large number of entities and the substantial interference due to the superposition. The key solution is the quasi-orthogonality since, in fact, with improved quasi-orthogonality of randomly initialized distributed representations, the entities become more distinguishable from their representations even after slight distortion, and the interference can be reduced, allowing more associations to be encoded into a single memory trace. In our recent work [68], we observed that by elementwise

sampling the initial representations from a heavy-tailed distribution, e.g., Cauchy distribution, the quasi-orthogonality of sampled random vectors could be significantly improved, realizing a dramatically enhanced memory capacity for associations.

An asymptotic approximation of the density function $g(\rho_C)$ was derived using the arithmetic of random variables and the generalized central limit theorem [34]. Let $\mathbf{X}_1, \dots, \mathbf{X}_n$ be independent q -dimensional random vectors that are elementwise i.i.d. sampled from a Cauchy distribution $\mathcal{C}(0, 1)$. Let Θ_{ij} denote the angle between vectors \mathbf{X}_i and \mathbf{X}_j and let $\rho_{ij} := \cos \Theta_{ij} \in [-1, 1]$. In the limit $q \rightarrow \infty$, the distribution function of the pairwise angle in the Cauchy case approaches

$$g(\rho_C) = -\frac{2}{\pi^2 q^2 \rho_C^3} \cdot \frac{1}{z^{\frac{3}{2}}} \left[e^{\frac{1}{\pi z}} \text{Ei} \left(-\frac{1}{\pi z} \right) \right], \quad (1.2)$$

where $z := \frac{1}{q^2} \left(\frac{1}{\rho_C^2} - 1 \right)$, and the exponential integral $\text{Ei}(x)$ is defined as $\text{Ei}(x) = -\int_{-x}^{\infty} \frac{e^{-t}}{t} dt$. Figure 1.5.2 shows the empirical distribution of ρ_G in an ensemble of random vectors elementwise sampled from the normal distribution $\mathcal{N}(0, 1)$ compared with the distribution function given in Eq. 1.1; and the empirical distribution of ρ_C in an ensemble of random vectors elementwise sampled from the Cauchy distribution $\mathcal{C}(0, 1)$ compared with the analytical approximation provided in Eq. 1.2. In particular, a spike concentrated around $\rho = 0$ in the setting of Cauchy initialization indicates a significantly improved quasi-orthogonality.

1.5.3 Holistic Representation

In the holographic reduced representation, circular convolution is employed as the reversible bind operation. Given two q -dimensional vectors \mathbf{a} and \mathbf{b} , the circular convolution $*$: $\mathbb{R}^q \times \mathbb{R}^q \rightarrow \mathbb{R}^q$ is defined as

$$[\mathbf{a} * \mathbf{b}]_k = \sum_{i=0}^{q-1} a_i b_{(k-i) \bmod q}.$$

Moreover, the circular correlation operator $\star : \mathbb{R}^q \times \mathbb{R}^q \rightarrow \mathbb{R}^q$ is defined as

$$[\mathbf{a} \star \mathbf{b}]_k = \sum_{i=0}^{q-1} a_i b_{(k+i) \bmod q}.$$

A noisy version of \mathbf{b} can be decoded from the association $\mathbf{a} * \mathbf{b}$ via the circular correlation, i.e., $\mathbf{b} \approx \mathbf{a} \star (\mathbf{a} * \mathbf{b})$. Figure 1.7 provides an illustrative explanation of circular convolution and circular correlation. By applying the convolution-correlation binding method to knowledge graphs, the memory trace of an entity should encode all the semantic triples

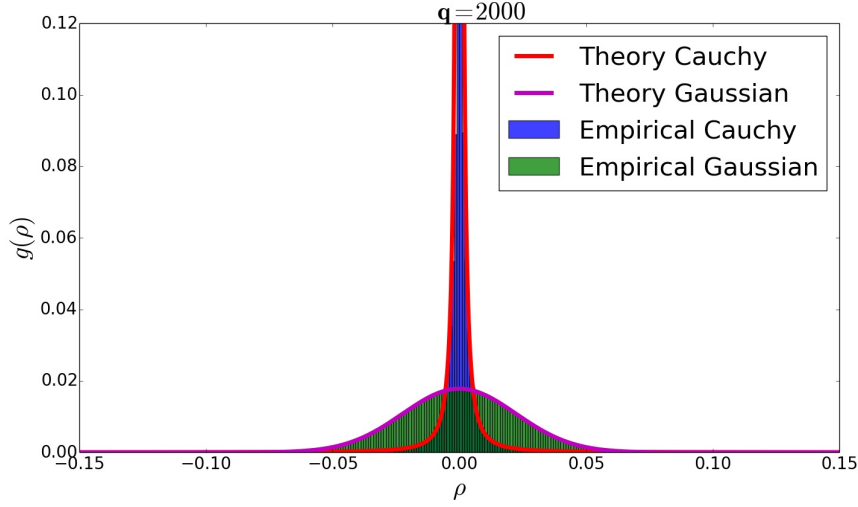


Figure 1.6: The empirical distribution of ρ_G in an ensemble of random vectors elementwise sampled from $\mathcal{N}(0, 1)$ (green) is compared with the asymptotic distribution function given in Eq. 1.1 (magenta); the empirical distribution of ρ_G in an ensemble of random vectors elementwise sampled from $\mathcal{C}(0, 1)$ (blue) is compared with its analytical approximation provided in Eq. 1.2 (red). In both cases, all the random vectors have dimension $q = 2000$.

that are related to the entity. For instance, suppose that (s, p_1, o_1) , (s, p_2, o_2) , and (s, p_3, o_3) are all the semantic triples that have s as the subject. Then the memory trace for s should encode all the predicate-object pairs that are associated with it, such that given the memory trace of s and a predicate, say p_1 or p_2 , as the cue, the object, o_1 or o_2 , should be retrieved.

The resulting memory traces are referred to as the holistic representations in [68] to emphasize the holistic theory of meaning. Note that the holistic representations of entities are *not* learned as in the statistical relational modeling of knowledge graph. Instead, they are encoded from random initializations. [68] also first mathematically proves that circular correlation is more suitable for encoding asymmetric relations in knowledge graphs, such as $(California, locatedIn, USA)$ for which the inverse relation $(USA, locatedIn, California)$ does not exist, since circular correlation is a noncommutative operator, i.e., $\mathbf{a} \star \mathbf{b} \neq \mathbf{b} \star \mathbf{a}$. In this case, circular convolution will be employed accordingly as the decoding operation.

Holistic representations realize an associative memory architecture by compressing each node's neighboring information into its representation. Each node could approximately encode its local structure in the knowledge graph. Therefore, it is expected that by employing

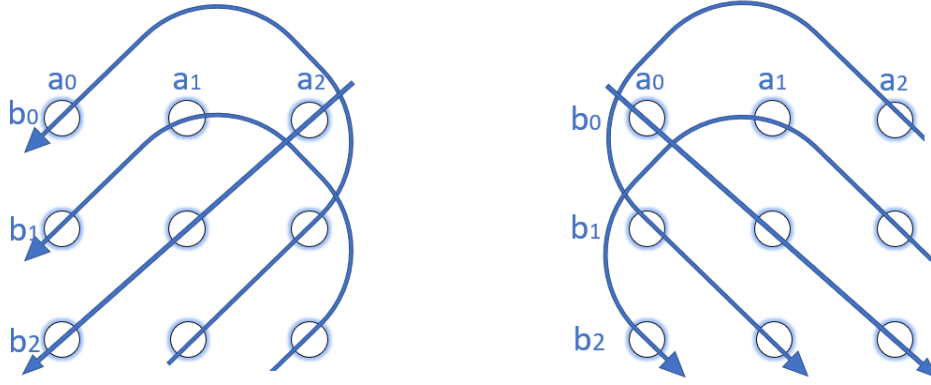


Figure 1.7: Illustration for circular convolution $[\mathbf{a} * \mathbf{b}]$ (left) and circular correlation $[\mathbf{a} \star \mathbf{b}]$ (right) of two 3-dimensional vectors \mathbf{a} and \mathbf{b} . For example, the vector after circular convolution reads $[\mathbf{a} * \mathbf{b}]_0 = a_0b_0 + a_2b_1 + a_1b_2$, $[\mathbf{a} * \mathbf{b}]_1 = a_1b_0 + a_0b_1 + a_2b_2$, and $[\mathbf{a} * \mathbf{b}]_2 = a_2b_0 + a_1b_1 + a_0b_2$. The vector after circular correlation reads $[\mathbf{a} \star \mathbf{b}]_0 = a_0b_0 + a_1b_1 + a_2b_2$, $[\mathbf{a} \star \mathbf{b}]_1 = a_2b_0 + a_0b_1 + a_1b_2$, and $[\mathbf{a} \star \mathbf{b}]_2 = a_1b_0 + a_2b_1 + a_0b_2$.

a global learning module, missing links, or implicit knowledge, in the knowledge graph can be inferred as well. In [68], we adopt a simple 2-layer neural network that uses the holistic representations as input features to learn the global relational patterns hidden in the knowledge graph. This simple neural network with bottle structure outperforms several baselines, especially when the holistic representations are encoded from the Cauchy initialization. More important, it is even capable of inferring implicit knowledge of unobserved entities given only several semantic triples that contain those unobserved entities, without retraining and fine-tuning the weights. More experiments and rigorous analysis of the holistic representations can be found in [68] and Chapter 3.

1.6 Variational Quantum Circuit for Knowledge Graph Embedding

1.6.1 Variational Quantum Circuit

Knowledge graphs are extracted from various unstructured text data, e.g., webpages, newspaper articles, and scientific reports, through two steps: named entity recognition and relation extraction. The task of named entity recognition (NER), also known as named entity classification, is to recognize and locate the mentioned entities in unstructured texts and

classify them to predefined categories [82]. Previous NER approaches use language-specific knowledge to design hand-crafted rules and annotate mentioned entities in corpora. With the development of deep learning-based natural language processing, neural architectures for NER are introduced in [60], which apply bidirectional LSTMs and conditional random fields. Relations are then extracted from the corpora after annotating the entities and integrated into knowledge bases as semantic triples.

The number of recognized entities and extracted triples continually increases as knowledge graphs collect and merge information from different data sources. The growing number of semantic triples and entities leads to a slow inference on knowledge graphs given a new query. To understand this, consider that we are given an unobserved query with an unknown object, the computational complexity of inferring the potentially correct object is $\mathcal{O}(N_e R^3)$ for the Tucker model, where R represents the rank, and N_e the number of entities. This estimation comes from the observation that the computational complexity of evaluating the score function is R^3 for the Tucker model, and the score function needs to be calculated N_e times and ranked afterward to determine the potential object. Hence, in this dissertation, we investigate the first quantum approaches for statistical relational learning to accelerate the learning process and inference on knowledge graphs. In this section, we briefly sketch the idea of our first quantum approach, whose underlying building block is the parameterized quantum circuit, also known as the variational quantum circuit.

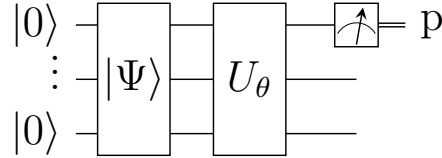


Figure 1.8: Quantum circuit part of the quantum-classical hybrid architecture for supervised learning. The input feature is first normalized and encoded as the amplitudes of the quantum state $|\Phi\rangle$, which is then evolved by a parameterized unitary transformation U_θ , where θ represents parameters in the transformation. The predicted binary label is encoded in the measurement statistics of the auxiliary qubit.

The parameterized quantum circuit is the building block of quantum-classical hybrid machine learning algorithms. Hybrid approaches combine a low-depth parameterized quantum circuit and a classical unit for optimization to learn a task by tuning and updating the parameters in the circuit. The hybrid architecture makes the parameterized quantum

circuit a hot research topic since it is more suited to near-term noisy quantum devices. An overview of variational algorithms with a quantum-classical hybrid optimization scheme can be found in [75, 78], which show that parameterized quantum circuits can approximate some nonlinear functions through numerical simulations. The variational approach can be applied to solve combinatorial optimization problems, such as MaxCut on regular graphs [26], by reformulating them to the ground state problems of Ising models. Moreover, [101] investigated a supervised learning algorithm using the quantum-classical hybrid architecture, where inputs are normalized and encoded into the amplitudes of quantum states.

Figure 1.8 illustrates the quantum part of the hybrid architecture for supervised learning. This quantum supervised learning architecture encodes the normalized input features as the amplitudes of a quantum state, which is then evolved by a sequence of unitary transformations. The unitary transformations are usually composed of parameterized single and two-qubit gates. An objective function for the binary classification is associated with the measurement statistics of an auxiliary qubit, which is entangled with the qubits for amplitude encoding. This binary quantum classifier is then optimized by updating the parameters in the unitary transformations and minimizing the objective function.

In this dissertation, we restrict to the case where the unitary transformation U_θ is composed of a sequence of parameterized single and two-qubit gates. A single-qubit gate is a 2-dimensional matrix representation of the special unitary group $SU(2)$, which, after ignoring a global phase, can be parameterized as

$$G(\alpha, \beta, \gamma) = \begin{pmatrix} e^{i\beta} \cos \alpha & e^{i\gamma} \sin \alpha \\ -e^{-i\gamma} \sin \alpha & e^{-i\beta} \cos \alpha \end{pmatrix}, \quad (1.3)$$

where $\{\alpha, \beta, \gamma\}$ are tunable parameters of the single-qubit gate. Two-qubit gates that we adopt in the Ansätze are controlled gates, where one qubit acts as a control of operations on another qubit. For instance, the controlled gate $C_i(G_j)$ that applies a unitary transformation on the j -th qubit conditioned on the state of the i -th qubit can be written as

$$C_i(G_j) |x\rangle_i \otimes |y\rangle_j = |x\rangle_i \otimes G_j^x |y\rangle_j,$$

where $|x\rangle_i$ and $|y\rangle_j$ represent the quantum state of qubit i and j , respectively.

Quantum algorithms using n fully entangled qubits can perform computations on 2^n amplitudes. Hence, an n -qubit quantum circuit can encode input data with maximal dimension 2^n . To understand how the circuit processes the input data, we explicitly write

down the $(2^n \times 2^n)$ -dimensional matrix representation of each unitary gate acting on the n -qubit system. Suppose that U_θ consists of L unitary operations and let the l -th unitary operation U_l be a single-qubit gate acting on the k -th qubit, then its matrix representation reads

$$U_l = \mathbb{1}_1 \otimes \cdots \otimes G_k \otimes \cdots \otimes \mathbb{1}_n.$$

If the l -th unitary operation is a controlled gate $C_i(G_j)$, which acts on the j -th qubit and conditioned on the i -th qubit, then U_l possesses the following matrix representation

$$\begin{aligned} U_l = & \mathbb{1}_1 \otimes \cdots \otimes \underbrace{\mathbb{P}_0}_{i\text{-th}} \otimes \cdots \otimes \underbrace{\mathbb{1}_j}_{j\text{-th}} \otimes \cdots \otimes \mathbb{1}_n \\ & + \mathbb{1}_1 \otimes \cdots \otimes \underbrace{\mathbb{P}_1}_{i\text{-th}} \otimes \cdots \otimes \underbrace{G_j}_{j\text{-th}} \otimes \cdots \otimes \mathbb{1}_n, \end{aligned}$$

where $\mathbb{P}_0 = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}$ and $\mathbb{P}_1 = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}$. Therefor, the $(2^n \times 2^n)$ -dimensional matrix representation of U_θ can be written as $U_\theta = U_L \cdots U_1$.

To optimize the circuit model, gradients of parameters can be estimated from the same circuit architecture using the parameter shift rule. This technique has been recently proposed in [39, 101, 27], and it shows that the partial derivative of the expectation of a quantum observable with respect to a circuit parameter can be decomposed into a sum of unitary operators. Hence, the partial derivatives with respect to the circuit parameters can be derived from the measurement statistics of the same auxiliary qubit using the parameter shift rule. For instance, let us consider parameterized single-qubit gate $G(\alpha, \beta, \gamma)$, whose partial derivatives with respect to α , β , and γ read

$$\begin{aligned} \frac{\partial}{\partial \alpha} G(\alpha, \beta, \gamma) &= G(\alpha + \frac{\pi}{2}, \beta, \gamma) \\ \frac{\partial}{\partial \beta} G(\alpha, \beta, \gamma) &= \frac{1}{2} G(\alpha, \beta + \frac{\pi}{2}, 0) + \frac{1}{2} G(\alpha, \beta + \frac{\pi}{2}, \pi) \\ \frac{\partial}{\partial \gamma} G(\alpha, \beta, \gamma) &= \frac{1}{2} G(\alpha, 0, \gamma + \frac{\pi}{2}) + \frac{1}{2} G(\alpha, \pi, \gamma + \frac{\pi}{2}). \end{aligned}$$

1.6.2 Modeling Knowledge Graphs with Variational Quantum Circuit

Having the knowledge of variational quantum circuits, we can introduce quantum embedding models for knowledge graphs. In the pioneering work [70], we contribute two different quantum embedding models: QCE and fQCE. In both models, each entity possesses

a quantum representation, which is encoded as the amplitudes of a quantum states. The only difference is that how these quantum representations of entities are prepared or loaded as the amplitudes of quantum states. In the QCE model, quantum representations are stored in a tree-structured classical memory, which can be accessed by a quantum algorithm to load the representations as quantum representations. This memory structure (see Figure 1.9) is a special Quantum Random Access Memory (QRAM) [33], which allows the vector representations to be loaded with exponential acceleration in the vector dimension. Since the QCE model is training-based, we have shown that the iterative parameter updates might ruin the exponential speedup gained during the preparation of quantum states. Hence, we were motivated to propose the fully-parameterized Quantum Circuit Embedding (FQCE).

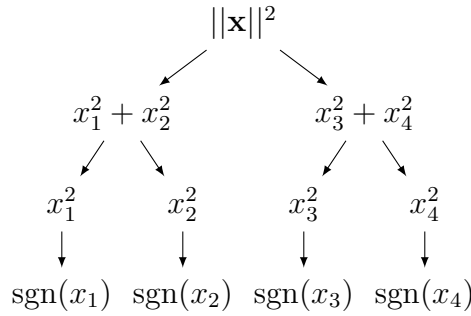


Figure 1.9: Classical memory structure with quantum access for creating the quantum state $|x\rangle = x_1|00\rangle + x_2|01\rangle + x_3|10\rangle + x_4|11\rangle$. In this example, a 4-dimensional real-valued normalized vector can be encoded as the amplitudes of a 2-qubit quantum state via three conditioned unitary rotations. In general, an R -dimensional real-valued vector can be encoded as the amplitudes of a $\lceil \log R \rceil$ -qubit quantum state via $\mathcal{O}(\log R)$ conditioned unitary rotations. More details are given in [90, 54]

In the FQCE model, vector representations are not stored in the classical memory structure described in Figure 1.9. Instead, quantum representations are prepared via additional variational quantum circuits with entity-dependent gate parameters. The entity-dependency means that the quantum circuit architecture for preparing entity quantum representations remains the same for all entities. However, each entity possesses a unique set of gate parameters. In other words, the quantum representation of an entity is prepared by iteratively applying parameterized gates on a maximally entangled state.

To evaluate the score function of a semantic triple, for both models, after preparing a quantum state for the subject, denoted as $|s\rangle$, a predicate-dependent circuit evolves the

quantum state $|s\rangle$ to the resulting state $|sp\rangle$. Moreover, the quantum state for the object $|o\rangle$ is prepared analogously, which is entangled with the state $|sp\rangle$ via an auxiliary qubit. After performing another Hadamard gate on the auxiliary qubit, the inner product of quantum states $|o\rangle$ and $|sp\rangle$ is encoded in the state of the auxiliary qubit. Therefore, we can derive the score function from the measurement statistics of the auxiliary qubit. More details of the circuit architecture can be found in [70] and Chapter 4.

By replacing the tree-structured memory storage with a variational quantum circuit for preparing the quantum representations, we realize a circuit-centric model for knowledge graph embedding. Recall that an R -dimensional classical vector representation can be encoded as the amplitudes of a quantum state with $\mathcal{O}(\log R)$ fully entangled qubits. Therefore, in the circuit-centric FQCE model, if the variational circuit for the entity preparation is shallow enough and in the order $\mathcal{O}(\log R)$, the computational complexity of score functions can be reduced to $\mathcal{O}(\log R)$.

Furthermore, we can realize an acceleration with respect to the number of entities when inferring unobserved triples after training. The basic idea is to introduce a quantum register for indices, which is entangled with the qubits for encoding quantum representations and the auxiliary qubit. Consider query $(s, p, ?)$ with an unknown object. Using the quantum register for indices, we first prepare states $|sp\rangle$ and $\sum_i |i\rangle |e_i\rangle$, where $\sum_i |i\rangle |e_i\rangle$ represents the entanglement of all indices of entities and the corresponding quantum representations. In this way, the inner product between $|sp\rangle$ and all $|e_i\rangle$ can be evaluated and encoded as the amplitudes of the register qubits. Correct objects might subsequently be read out by measuring the register qubits. This algorithm heuristically realizes a quadratic speedup in the number of entities during the inference. More details about the algorithm can be found in Section 6 of [70] and Chapter 4.

1.7 Quantum Tensor SVD for Knowledge Graphs Inference

1.7.1 Classical Tensor Singular Value Decomposition

In the last section, we have introduced one quantum approach for modeling knowledge graphs, which provides a quadratic acceleration for the knowledge inference with respect to the number of entities. This approach applies variational quantum circuits, realizing a learning-based method, and the inference is performed using the learned quantum

representations. The computational complexity of the learning-based approach is always proportional to the number of data points in the training set. Due to the vast number of semantic triples, it is still challenging to scale to more massive knowledge graphs. However, the tensor perspective of knowledge graphs indicates that the quantum counterparts of classical tensor decomposition algorithms for knowledge inference might have different computational complexities. In this section, we propose a sampling-based quantum method that realizes an exponential acceleration in knowledge reasoning.

Quantum machine learning (QML) has attracted the attention of scientists from different research areas since it has been shown that specific classical algorithms can be accelerated using quantum subroutines implemented on quantum devices. A famous quantum machine learning algorithm in recent years has been the quantum algorithm for linear systems of equations [43], also known as the HHL algorithm. The HHL algorithm performs a matrix inversion and offers an exponential speedup in the dimensions of the matrix under certain conditions. This quantum routine for matrix inversion finds applications in accelerating classical algorithms, e.g., in support vector machine for classification [92] and in linear regression [122]. Another series of quantum machine learning algorithms employ amplitude amplification [15] to solve supervised and clustering problems [123, 3]. The amplitude amplification quantum routine is inspired by Grover's database search algorithm [38] and can provide quadratic speedup to the problems mentioned above.

An interesting application of quantum machine learning is the quantum recommendation system [54]. Given the preference matrix, the quantum recommendation system can recommend user-preferred items with runtime polylogarithmic in the dimensions of the preference matrix. The quantum recommendation system relies on quantum singular value estimation, which requires an efficient encoding of the preference matrix into a quantum state. This quantum state preparation can be realized by the Quantum Random Access Memory described in Figure 1.9 with the entries of the preference matrix stored in the tree-structured memory. Recently, a quantum-inspired classical algorithm proposed in [107] suggests that there exists a classical algorithm that can achieve similar exponential acceleration if the classical algorithm can access and prepare the data with runtime similar to QRAM. However, as pointed out in [53], this quantum-inspired dequantization algorithm has a much higher polynomial dependence on the rank of the preference matrix as well as the inverse accuracy parameter, making this dequantized algorithm impractical, and the supremacy of the quantum recommendation system maintains.

Due to the three-way tensor view of semantic knowledge graphs, we propose a quantum

counterpart of classical tensor singular value decomposition, also called quantum tensor SVD. Inferring implicit semantic knowledge from observed samples is, in principal, a tensor completion problem. Therefore, before developing the quantum counterpart, it is necessary to prove the plausibility of classical tensor singular value decomposition for the tensor completion task. Notably, it is essential to determine under what conditions the initial tensor can be approximately reconstructed from the partially observed entries using the tensor SVD algorithm. Such conditions are crucial to select the appropriate quantum subroutines and design the quantum algorithm. The primary assumption made for the feasibility of classical tensor SVD algorithm is that the original knowledge graph contains global and structured relational patterns, such that low-rank approximation of the subsampled knowledge graph can approximately reconstruct the original one. In the following, we explain tensor SVD and its approximation power using rigorous mathematical language.

Let $\mathcal{A} = (\mathcal{A}_{i_1 i_2 \dots i_N}) \in \mathbb{R}^{d_1 \times d_2 \times \dots \times d_N}$ denote a N -way tensor with d_k representing the k -th dimension. The tensor product of two N -way tensors \mathcal{A} and \mathcal{B} is defined as $\langle \mathcal{A}, \mathcal{B} \rangle_F := \sum_{i_1=1}^{d_1} \dots \sum_{i_N=1}^{d_N} \mathcal{A}_{i_1 i_2 \dots i_N} \mathcal{B}_{i_1 i_2 \dots i_N}$. The tensor-vector product can be written as

$$\mathcal{A} \otimes_1 \mathbf{x}_1 \cdots \otimes_N \mathbf{x}_N := \sum_{i_1=1}^{d_1} \cdots \sum_{i_N=1}^{d_N} \mathcal{A}_{i_1 i_2 \dots i_N} x_{1i_1} x_{2i_2} \cdots x_{Ni_N},$$

for arbitrary vectors $\mathbf{x}_k \in \mathbb{R}^{d_k}$, with $k = 1, \dots, N$. We introduce two tensor norms: the Frobenius norm and the spectral norm. The Frobenius norm of tensor \mathcal{A} reads $\|\mathcal{A}\|_F := \sqrt{\langle \mathcal{A}, \mathcal{A} \rangle_F}$. The spectral norm $\|\mathcal{A}\|_\sigma$ is defined as

$$\|\mathcal{A}\|_\sigma = \max\{\mathcal{A} \otimes_1 \mathbf{x}_1 \cdots \otimes_N \mathbf{x}_N | \mathbf{x}_k \in S^{d_k-1}, k = 1, \dots, N\},$$

where S^{d_k-1} represents a unit vector in \mathbb{R}^{d_k} .

Following the work [19], the definition of classical tensor SVD is provided below.

Definition 2 (cf. Definition 1 in [72]). *If a tensor $\mathcal{A} \in \mathbb{R}^{d_1 \times d_2 \times \dots \times d_N}$ can be written as sum of rank-1 outer product tensors $\mathcal{A} = \sum_{i=1}^R \sigma_i u_1^{(i)} \otimes u_2^{(i)} \cdots \otimes u_N^{(i)}$, with singular values $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_R$ and $\langle u_k^{(i)}, u_k^{(j)} \rangle = \delta_{ij}$, for $k = 1, \dots, N$. Then we say \mathcal{A} has a tensor singular value decomposition with rank R .*

The above notations and tensor SVD are defined for general N -way tensors. We return to the 3-way semantic tensor χ representing a semantic knowledge graph for a moment. It was first proved in [72] by us that if a low-rank factorization $\tilde{\chi}$ can well approximate the original semantic tensor χ , i.e., $\|\chi - \tilde{\chi}\|_F \leq \epsilon \|\chi\|_F$ for a small error $\epsilon > 0$, then information

can be successfully retrieved with high probability. To be more specific, suppose that we use the approximation tensor $\tilde{\chi}$ to answer the query $(s, p, ?)$ for an unknown object. The probability that correct objects cannot be successfully located from the top- n returns provided by $\tilde{\chi}$ is bounded by $\mathcal{O}(\epsilon^n)$ for sufficiently small error ϵ (cf. Lemma 1 in [72]). Note that, in general, the semantic tensor χ is incomplete with missing entries, or subsampled, and the task is to infer unobserved entries from the approximation of subsampled tensor. Therefore, it is essential to understand under what circumstances a tensor can be well approximated by the low-rank factorization of its subsampled tensor.

Let us go back to the general tensor \mathcal{A} . We let $\hat{\mathcal{A}}$ denote the subsampled tensor and introduce the subsampling and rescaling scheme suggested in [2], where

$$\hat{\mathcal{A}}_{i_1 i_2 \dots i_N} = \begin{cases} \frac{\mathcal{A}_{i_1 i_2 \dots i_N}}{p} & \text{with probability } p \\ 0 & \text{otherwise.} \end{cases}$$

This procedure describes that tensor elements are first i.i.d. subsampled with probability p and subsequently rescaled. The advantage of rescaling is that expectation values of elements in the subsampled tensor remain the same as before the subsampling. We can write the subsampled tensor as $\hat{\mathcal{A}} = \mathcal{A} + \mathcal{N}$, where the tensor \mathcal{N} represents introduced noise after subsampling. In fact, to prove that the reconstruction error from the low-rank approximation of the subsampled tensor, we need to bound the spectral and Frobenius norms of the noise tensor \mathcal{N} , which can be considered as our contributions to the theory of tensor decomposition.

We apply tensor factorization and subspace projection to the subsampled and rescaled tensor $\hat{\mathcal{A}}$ to approximate the original tensor \mathcal{A} . The idea behind these operations is that after projecting $\hat{\mathcal{A}}$ to low-rank subspaces, observed elements become smoothed, and missing entries are boosted, such that missing entries, or implicit knowledge, can be retrieved from the reconstruction. In particular, we provide two subspace projection methods after tensor SVD. The first method project the decomposed tensor onto the subspaces spanned by the top- r singular values, which is also referred to as the truncated r -rank tensor SVD, denoted as $\hat{\mathcal{A}}_r$. The second approach projects the factorized tensor onto the subspaces spanned by the singular values whose absolute values are larger than a cutoff threshold $\tau > 0$. We let $\hat{\mathcal{A}}_{|\cdot| \geq \tau}$ denote the second projection method and name it as the projected tensor SVD with absolute singular value threshold τ .

In [72], as one significant theoretical contribution, we prove that if the original tensor \mathcal{A} possesses a low-rank approximation, then the reconstruction error of truncated r -rank ten-

sor SVD using the subsampled and rescaled tensor $\hat{\mathcal{A}}$ is bounded, i.e., $\|\mathcal{A} - \hat{\mathcal{A}}_r\|_F \leq \epsilon \|\mathcal{A}\|_F$ with high probability, where $\epsilon > 0$ is a small number depending on the subsample probability (cf. Theorem 1 in [72]). However, it could be challenging to design the corresponding quantum algorithm of truncated low-rank tensor SVD. The reason is that negative singular values might occur in the tensor case, and quantum subroutines for singular values always disregard the sign of singular values. To be more specific, the quantum subroutines that we need, such as quantum singular value estimation and singular value projection, neglect the sign of singular values and store the absolute values of them in quantum registers. Therefore, we are motivated to propose the classical algorithm of projected tensor SVD with threshold and design the corresponding quantum counterpart of it. Furthermore, we show that the tensor reconstruction error using the second approach is also bounded, i.e., $\|\mathcal{A} - \hat{\mathcal{A}}_{|\cdot| \geq \tau}\|_F \leq \epsilon \|\mathcal{A}\|_F$ is satisfied with high probability (cf. Theorem 2 in [72]).

1.7.2 Sampling-based Quantum Algorithm for Knowledge Graphs Inference

The quantum algorithm for tensor completion is then built on the classical algorithm of projected tensor SVD with a projection threshold. We briefly sketch the idea of quantum tensor SVD in this paragraph; more details are relegated to Section 3 in [72] and Chapter 5. Consider we are given query $(s, p, ?)$ for correct objects. The sampling-based quantum algorithm should sample semantic triples with given subject s and subsequently post-select on the predicate p . After quantum sampling and post-selection, we obtain the desired semantic triples to identify correct objects. In other words, sampling should be conducted in the predicate and object dimensions, which can be realized by unfolding the subsampled semantic tensor $\hat{\chi}$. Therefore, as an essential step, we need to prepare the following quantum density operator from $\hat{\chi}$,

$$\rho_{\hat{\chi}^\dagger \hat{\chi}} := \sum_{i_2 i_3 i'_2 i'_3} \sum_{i_1} \hat{\chi}_{i_1, i_2 i_3}^\dagger \hat{\chi}_{i_1, i'_2 i'_3} |i_2 i_3\rangle \langle i'_2 i'_3|.$$

This can be done by encoding the tensor $\hat{\chi}$ into the quantum state $\sum_{i_1 i_2 i_3} \hat{\chi}_{i_1 i_2 i_3} |i_1 i_2 i_3\rangle$ and subsequently performing a partial trace to the density function of this quantum state. In order to maintain the quantum supremacy, the quantum state $\sum_{i_1 i_2 i_3} \hat{\chi}_{i_1 i_2 i_3} |i_1 i_2 i_3\rangle$ and the operator $\rho_{\hat{\chi}^\dagger \hat{\chi}}$ should be prepared using QRAM, which realizes an exponential acceleration in the tensor dimensions.

After obtaining the density operator $\rho_{\hat{\chi}^\dagger \hat{\chi}}$, the next step is to project this operator onto eigenspaces spanned by the eigenvalues whose absolute values are larger than a threshold τ . To realize the projection, we first need to exponentiate the density matrix and perform the quantum phase estimation [57], such that eigenvalues of $\rho_{\hat{\chi}^\dagger \hat{\chi}}$ are all stored in additional quantum registers. This step is also known as the quantum principal component analysis (qPCA) [67]. Afterward, analogous to the quantum projection introduced in [54], we can achieve a quantum singular value projection with selected eigenvalues larger than the threshold τ with the help of an auxiliary quantum register. The resulting quantum state is related to the classical projected tensor $\hat{\chi}_{|\cdot| \geq \tau}$, where values of missing entries become boosted. Therefore, correct objects might be sampled from the resulting quantum states by measuring the canonical bases. The computational complexity for inferring the query $(s, p, ?)$ is, therefore, $\mathcal{O}(\text{polylog}(d_1 d_2 d_3))$, where d_1 , d_2 , and d_3 are three dimensions of the semantic tensor, realizing an exponential acceleration during the inference. More details of the analysis of the quantum tensor SVD can be found in Section 3 of [72] and Chapter 5.

1.8 Causal Inference under Interference

1.8.1 Introduction

Determination of the causal connection between two time-independent variables from observational data is a major topic in causal inference. For instance, genetics studies genetic variants to find the causation of a disease, and sociology studies how education affects the income. Causal inference analyzes the potential outcome of the effect variable after changing the cause variable, also known as counterfactual inference [88, 79], which distinguishes it from the inference of correlation. Correlations between genetic variants and diseases only characterize the dependency between them, while the causation emphasizes that the presence of some genetic variants will lead to a specific disease, not the other way around. Hence, correlation does not always imply causation.

The framework that analyzes the cause and effect from observational or experimental studies to infer potential outcomes is called the Rubin causal model, also known as the Neyman-Rubin causal model [95, 97]. Estimating average treatment effects or individual treatment effects from observational or experimental studies are essential tasks in the Rubin causal model. Causal inference usually requires the Stable Unit Treatment Value Assumption (SUTVA) [20, 96], which assumes that the response of one unit is consistently

observed and should be unaffected by the treatment assignments and responses of other units. Namely, the causal model should be free from the interference between units. The interference-free assumption becomes problematic in the relational setting, e.g., under a social network setting, since the response of a unit might be affected by its social neighbors through peer effects.

Causal inference in the complex relational domain is a challenging yet intriguing research topic. For example, let us consider a citation knowledge graph of scientific publications, where nodes represent authors, publications, and journals, and labeled edges represent the friendships or mentorships between authors, as well as authored-by relations. A causal model can discover that the co-authorship of a publication might affect its citation, and the topic of a paper might cause where it can be published [74]. In this dissertation, we focus on the Rubin causal model, and for the sake of simplicity, we investigate causal inference under interference on networks without labeled edges. Possible scenarios are the following: an individual’s health condition might be dependent on its social connections’ vaccination conditions against an infectious disease, or other persons might influence the inclination of a person buying a particular product through opinion propagation on social networks.

Causal inference with interference was first studied in [50], which provides estimators for group-level causal effects randomized trials, and the interference is presumed to appear only within the groups. Later, based on this work, [108] proposes new inverse probability weighting estimators for finite sample causal inference with a binary outcome, and it uses observed data from group-randomized experiments in the presence of interference. Furthermore, [66] derives the asymptotic causal estimators when either the number of subjects per group or the number of groups diverges. Group-level randomized experiments and partial interference within the groups and independence across different groups are, however, sometimes invalid assumptions. Hence, several works focus on unit-level causal effects under cross-unit interference and arbitrary treatment assignments, such as [4, 29, 86, 87, 119]. For instance, [4] performs causal inference via the Horvitz-Thompson estimator, assuming that the vector of the generalized probability of exposure for each unit is known. [29] develops new covariate-adjustment methods for causal inference on networks based on neighborhood propensity score and individual propensity score. [87] fits generalized linear models for estimating non-instantaneous contagion and infectiousness effects in a social network.

We first introduce notations for the Rubin causal inference model. Let the binary

variable T_i denote the treatment assignment of node or unit i with $T_i = 1$ indicating that node i is assigned to the treatment group, and $T_i = 0$ if node i is in the control group. Moreover, let \mathbf{X}_i be the covariate vector of node i and Y_i the outcome variable. Note that in the interference-free assumption, the outcome variable Y_i might only depend on the assignment T_i and the covariate \mathbf{X}_i . Hence, we let $Y_i(T_i = 1)$ represent the response under treatment and $Y_i(T_i = 0)$ the potential response under control. Individual treatment effect (IDE) of node i is then defined as the difference between responses under treatment and control, i.e.,

$$\tau(\mathbf{X}_i) := \mathbb{E}[Y_i(T_i = 1) - Y_i(T_i = 0) | \mathbf{X}_i],$$

where the expectation value is taken over the response variable. One approach for estimating the individual treatment effect is directly modeling the data from random experiments or observational studies. Given n data points ² $(\mathbf{X}_i, T_i, Y_i^F)$ with the factual response $Y_i^F := T_i Y_i(T_i = 1) + (1 - T_i) Y_i(T_i = 0)$, we learn a causal estimator h such that $h(\mathbf{X}_i, T_i) \approx Y_i^F$. Then, the estimated individual treatment effect can be derived as

$$\hat{\tau}(\mathbf{X}_i) = \begin{cases} Y_i^F - h(\mathbf{X}_i, T_i = 0), & T_i = 1, \\ h(\mathbf{X}_i, T_i = 1) - Y_i^F, & T_i = 0, \end{cases}$$

where evaluating the unobserved responses is also known as counterfactual inference.

In order to estimate the individual treatment effect, we need to evaluate counterfactual outcomes using obtained causal estimators, which makes this estimation method insufficient, especially when the treatment and control groups are imbalanced. Imbalanced treatment and control groups can cause biased causal estimators. Previous approaches for addressing the imbalance issue are, e.g., propensity score matching and propensity score reweighting [94, 93, 7]. Besides, we always encounter the covariate shift or domain adaption problem when the factual distribution $\Pr(\mathbf{X}_i, T_i)$ differs from the counterfactual distribution $\Pr(\mathbf{X}_i, 1 - T_i)$. This problem is prevalent in causal inference with observed data, where treatments are not randomly assigned. To address the issues mentioned above, [52, 102] propose a method for learning balanced representations. In this method, covariate vectors are first mapped to a feature space via a feature map Φ , such that in this feature space, the discrepancy between the empirical distribution $\hat{\Pr}(\Phi(\mathbf{X}_i), T_i)$ and the empirical counterfactual distribution $\hat{\Pr}(\Phi(\mathbf{X}_i), 1 - T_i)$ becomes minimized (see Figure 1.10). To some extent, the treatment assignment T_i and the representation $\Phi(\mathbf{X}_i)$ in the feature space become approximately disentangled, i.e., $\Pr(\mathbf{X}_i, T_i) \approx \Pr(\mathbf{X}_i) \Pr(T_i)$.

²For the sake of simplicity, we abuse the notation and let capital letters represent values of variables.

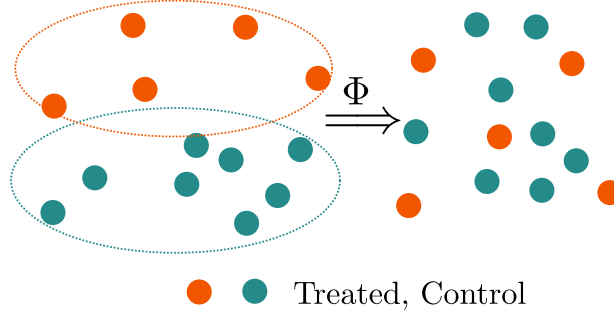


Figure 1.10: Illustration of the idea of learning balanced representation. In the feature space, representations $\Phi(\mathbf{X}_i)$ become independent on the treatment assignments T_i after an imbalance penalty.

Now we introduce the setting of causal inference under network interference. Let $\mathcal{G} = (\mathcal{N}, \mathcal{E}, A)$ be a graph network with node set \mathcal{N} of size n , edge set \mathcal{E} , and adjacency matrix $A \in \{0, 1\}^{n \times n}$. We use \mathcal{N}_i to represent the set of neighboring nodes of a node $i \in \mathcal{N}$. Let $T_{\mathcal{N}_i}$ and $Y_{\mathcal{N}_i}$ indicate the treatment assignments and potential responses of neighboring nodes \mathcal{N}_i of the node i . Moreover, let \mathbf{X} and \mathbf{T} represent the covariate vectors and the treatment assignments of all nodes, respectively. The idea of the Rubin causal model is to estimate the population-level causal effects from randomized experiments, where individuals are assigned randomly to the treatment and control groups. However, in many circumstances, randomized assignments are infeasible due to external factors, budget constraints, or the nonexperimental setting.

In this dissertation, we study causal inference under network interference using data in both experimental and observational settings. We use the following structural equation model which describes the data generation process to unify both experimental and observational settings,

$$\begin{aligned}
 T_i &= f_T(X_i), \\
 Y_i &= f_Y(T_i, \mathbf{X}, \mathbf{T}, \mathcal{G}) + \epsilon_{Y_i},
 \end{aligned} \tag{1.4}$$

for units $i = 1, \dots, n$, where f_T captures the assignment mechanism, and f_Y describes the simulation mechanism of potential outcomes. In the randomized experiment setting, the assignment mechanism assumes that each individual is assigned to the treatment with a predefined treatment probability p , i.e., $f_T = \text{Bern}(p)$. However, in the setting of observational studies, we model the assignment mechanism using covariates. We use a function f_Y to simulate the causal responses under general network interference, which is a function not

only of \mathbf{X}_i and \mathbf{T}_i but also the network structure and neighboring treatment assignments and covariates.

Various methods for estimating individual treatment and spillover effects under network interference have been proposed. [110, 14, 4] introduce an exposure variable G_i as a function of neighboring treatment assignments $T_{\mathcal{N}_i}$. One choice for G_i could be the exposure level to the treated neighbors, i.e., $G_i := \frac{\sum_{j \in \mathcal{N}_i} T_j}{|\mathcal{N}_i|}$. Furthermore, [29] proves the identifiability of individual treatment effect under the assumption that potential response only depends on the individual treatment assignment and the level of exposure. [29] further defines an individual treatment effect under the exposure $G_i = g$,

$$\tau(\mathbf{X}_i, G_i = g) := \mathbb{E}[Y_i(T_i = 1, G_i = g) - Y_i(T_i = 0, G_i = g) | \mathbf{X}_i]. \quad (1.5)$$

In addition, the spillover effect received by node i under the treatment assignment $T_i = t$ and the exposure $G_i = g$ is defined as

$$\delta(\mathbf{X}_i, T_i = t, G_i = g) := \mathbb{E}[Y_i(T_i = t, G_i = g) - Y_i(T_i = t, G_i = 0) | \mathbf{X}_i].$$

Parametric causal estimators with generalized propensity score weighting are employed to estimate the individual treatment and spillover effects.

In most circumstances, the assumption made in [29] becomes insufficient since the response could be a complicated function of the network structure, neighboring covariates, and treatment assignments. [86] investigates the estimation and inference of causal effects in the social network setting and assumes a more general causal structural model. In that causal structural model, the treatment response of node i under assignment T_i is a function of its covariate, neighboring covariates, and treatment assignments, i.e.,

$$Y_{i,t} := f_Y(s_X(\mathbf{X}_i, \{\mathbf{X}_j | j \in \mathcal{N}_i\}), s_T(T_i, \{T_j | j \in \mathcal{N}_i\})).$$

The summary functions s_X and s_T could be, for instance, the concatenations of covariates and assignments, which are defined as $s_X(\mathbf{X}_i, \{\mathbf{X}_j | j \in \mathcal{N}_i\}) := (\mathbf{X}_i, \sum_{j \in \mathcal{N}_i} \mathbf{X}_j)$ and $s_T(T_i, \{T_j | j \in \mathcal{N}_i\}) := (T_i, \sum_{j \in \mathcal{N}_i} T_j)$, respectively. These summary functions might cause high-dimensional and high-variance inputs to the causal estimator, and more critical, they rule out the possibility of modeling network interference beyond nearest neighbors. Hence, in this dissertation, we are motivated to investigate causal estimators which incorporate Graph Neural Networks. These GNN-based causal estimators can aggregate neighboring and higher-order neighboring features and treatment assignments, which make them superior candidates for studying spillover effects.

1.8.2 GNN-based Causal Estimators

Before elaborating on our GNN-based causal estimators, we first introduce graph neural networks that are employed in the model. Graph neural networks were proposed in the pioneering works [35, 99], which are based on the information diffusion mechanism and can be applied to general graphs. Graph neural networks are also referred to as Graph Convolutional Networks (GCNs) since they generalize the convolution operation from grid-structured data to graph-structured data. Propagating a node’s neighboring representations and integrating aggregated information into the node’s representation are essential operations of GCNs. The resulting representation that integrates neighboring information can be further applied to the next layer of graph convolutional operation, such that the final representation of a node after several layers of graph convolutional operations can aggregate higher-order neighboring features. GCNs can extract both local and global information by stacking several local graph convolutional operations, making them appropriate candidates for node- and graph-level classifications [56, 126].

A representative set of graph neural networks are spectral-based GCNs [56, 23]. The basic idea of spectral-based GCNs is to factorize the normalized Laplacian matrix and perform graph Fourier transform to graph signals. By inheriting previously introduced notations, the normalized graph Laplacian of an undirected graph is defined as $\mathbf{L} = \mathbf{I}_n - \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$, where \mathbf{I}_n is the diagonal identity matrix, and \mathbf{D} represents the node degree matrix, i.e., $\mathbf{D}_{ii} = \sum_j \mathbf{A}_{ij}$. [56] proposes a normalization trick to stabilize the numerical calculation by introducing the convolutional operator $\hat{\mathbf{D}}^{-1/2}\hat{\mathbf{A}}\hat{\mathbf{D}}^{-1/2}$, where the modified adjacent matrix $\hat{\mathbf{A}}$ is augmented by self-loops, i.e., $\hat{\mathbf{A}} := \mathbf{I}_n + \mathbf{A}$, and $\hat{\mathbf{D}}$ represents the node degree matrix of $\hat{\mathbf{A}}$. The corresponding convolutional layer is then defined as

$$\mathbf{X}^{(l+1)} = \sigma \left(\hat{\mathbf{D}}^{-1/2} \hat{\mathbf{A}} \hat{\mathbf{D}}^{-1/2} \mathbf{X}^{(l)} \mathbf{W}^{(l)} \right),$$

where $\mathbf{X}^{(l)}$ is the l -th layer hidden representation with $\mathbf{X}^{(0)}$ indicating input features, and we use ReLU as the activation function σ .

Another representative set of graph neural networks contain spatial-based GNNs [6, 41]. The basic idea of spatial-based GNNs is to propagate messages along edges with processed information, e.g., by taking non-identical contributions from neighbors using an attention mechanism [118]. One spatial-based GNN employed in the GNN-based casual estimators is the GraphSAGE model [41], whose message passing operator reads

$$\mathbf{X}_i^{(l+1)} = \text{norm} \left(\text{mean}_{j \in \mathcal{N}_i \cup \{i\}} \mathbf{X}_j^{(l)} \mathbf{W}^{(l)} \right),$$

where mean represents the operation of taking the mean value, and norm is a normalization operation. In GraphSAGE, neighboring nodes' features are first transformed via a weight matrix before integrating into the central node's representation. As first pointed out in [71], this localized information aggregation resembles a specific simulation mechanism of spillover effects on the network. Hence, spatial-based GNNs are expected to be compelling candidates for estimating causal effects on the network. One variation of the GraphSAGE model that can be applied in the causal estimators is proposed in [80], which transforms the central node's representation and the neighboring representations separately with different weight matrices, whose message passing operator is defined as

$$\mathbf{X}_i^{(l+1)} = \sigma \left(\mathbf{X}_i^{(l)} \mathbf{W}_1^{(l)} + \text{mean}_{j \in \mathcal{N}_i} \mathbf{X}_j^{(l)} \mathbf{W}_2^{(l)} \right).$$

This model, also known as the 1-GNN model, is expected to be more expressive than GraphSAGE due to the separate transformations. A detailed survey of graph neural networks can be found in [124].

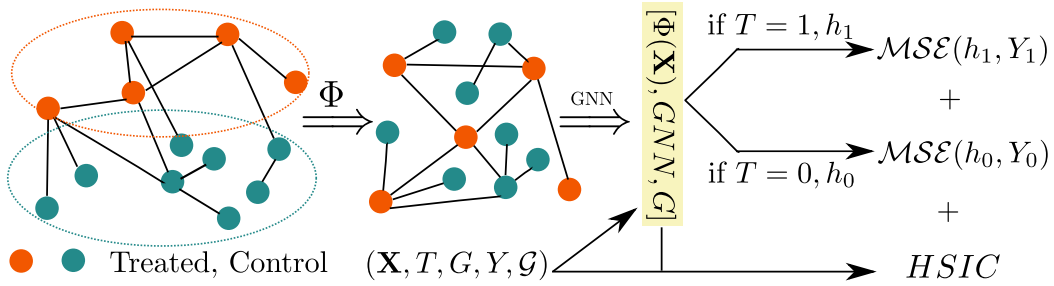


Figure 1.11: Illustration of GNN-based causal estimators.

Equipped with the necessary knowledge, we can now briefly sketch the GNN-based causal estimators (see Figure 1.11). GNN-based causal estimators consist mainly of four components: the feature map module Φ that maps the input covariates to a feature space; a graph neural network module that aggregates neighboring features and extracts local representations; an output module that uses the concatenations of central node feature and local representation as inputs to estimate the potential outcomes; and a representation penalty, with which node features and treatment assignments become disentangled in the feature space. We use the Hilbert-Schmidt Independence Criterion (HSIC) [37] to force the independence between features and treatment assignments. In addition to the balancing of the feature map's outputs, we also investigate the effect of balancing the outputs of the GNNs since one challenge of inferring causal effects under interference is the imbalanced spillover exposure.

One necessary clarification is that the outcome prediction networks h_0 and h_1 in Figure 1.11 can only estimate the causal effects that are the superpositions of individual treatment effects and network spillover effects. However, we can show that the individual treatment effects can be well extracted from GNN-based causal estimators by merely assuming that the considered unit is isolated and setting the graph as an empty graph without connections, i.e., $\mathcal{G} = \emptyset$. As a theoretical contribution in [71], we provide an error bound for the GNN-based causal estimators under reasonable assumptions. Particularly, we show that, if the maximal node degree is independent on the graph size n , the error is bounded by $\mathcal{O}(\sqrt{\frac{1}{n}})$. However, if the maximal node degree changes along with the graph size, the $\mathcal{O}(\sqrt{\frac{1}{n}})$ convergence rate becomes infeasible, which is in line with the theoretical observation reported in [86]. More details can be found in the Appendix of [71] and Chapter 6.

The fundamental problem of causal inference is the missing counterfactual responses since we cannot observe both outcomes of an individual under different assignments at once. Therefore we conduct randomized experiments on synthetic datasets with known response generation processes. In the setting of the randomized experiment, we choose two networks: an in-school friendship network collected from the National Longitudinal Study of Adolescent Health project [18] and an online social network in Slovakia [106]. We further test the proposed causal estimators in the observational setting using the Amazon dataset [65], where counterfactual responses are simulated by matching the covariates.

To thoroughly investigate the performance of GNN-based causal estimators, we consider both linear and nonlinear response generation mechanisms. For instance, the linear generation mechanism simulates the response as

$$Y_i = Y_i(T_i = 0, \mathcal{G} = \emptyset) + T_i \tau(\mathbf{X}_i) + \delta_i(\mathbf{X}, \mathbf{T}, \mathcal{G}) + \epsilon_{Y_i}, \quad (1.6)$$

where $Y_i(T_i = 0, \mathcal{G} = \emptyset)$ and $\tau(\mathbf{X}_i)$ represent response under control and individual treatment effect without network interference, respectively; $\delta_i(\mathbf{X}, \mathbf{T}, \mathcal{G})$ indicates the spillover effect; ϵ_{Y_i} is Gaussian noise. Note that $Y_i(T_i = 0, \mathcal{G} = \emptyset)$ and $\tau(\mathbf{X}_i)$ are nonlinear functions of covariates, and $\delta_i(\mathbf{X}, \mathbf{T}, \mathcal{G})$ is simulated as a function of individual treatment effects, treatment assignments, and network structure. Nonlinear responses are simulated analogously by, for example, including a quadratic term of $\delta_i(\mathbf{X}, \mathbf{T}, \mathcal{G})$. GNN-based causal estimators have shown superior performance on causal effects prediction and individual treatment effects retrieval for both linear and nonlinear responses.

1.8.3 Intervention Optimization on Network

Given causal estimators derived from randomized experiments or observational studies, an optimal intervention policy can be learned to maximize the total welfare of the population. In many circumstances, policymakers might face some specific constraints that the learned policy needs to satisfy, e.g., capacity or budget constraint. For instance, policymakers might need to choose whom to treat, such that a predefined upper limit bounds the percentage of treated individuals, and at the same time, the average welfare of the population can be maximized. Under the interference-free condition, [5] suggests using a utility function to study the average utility for the population by applying a treatment assignment policy π , which is defined as

$$A(\pi) = \mathbb{E}[(2\pi(\mathbf{X}_i) - 1)(Y_i(T_i = 1) - Y_i(T_i = 0))] = \mathbb{E}[(2\pi(\mathbf{X}_i) - 1)\tau(\mathbf{X}_i)].$$

Intuitively speaking, the policy learns to assign individuals with positive IDE to the treatment group and individuals with negative IDE to the control group. In practice, an optimal empirical policy $\hat{\pi}_n$ can be learned by maximizing the following empirical utility function from n samples

$$\hat{A}_n^\tau(\pi) := \frac{1}{n} \sum_{i=1}^n (2\pi(\mathbf{X}_i) - 1) \hat{\tau}(\mathbf{X}_i),$$

where the outcome estimator $\hat{\tau}$ is plugged in. [5] further establishes strong guarantees for policy regret, which quantify the utilitarian difference between optimal empirical policies and optimal global policies.

As another contribution, we define a utility function in the network setting. For the sake of notational simplicity, we consider only interference from first-order neighbors and write the response variable as $Y_i(T_i, \mathbf{X}_{\mathcal{N}_i}, T_{\mathcal{N}_i})$. The corresponding utility function of an intervention policy π on a network is then defined as

$$S(\pi) := \mathbb{E}[(2\pi(\mathbf{X}_i) - 1)(Y_i(T_i = 1, \mathbf{X}_{\mathcal{N}_i}, T_{\mathcal{N}_i} = \pi(\mathbf{X}_{\mathcal{N}_i})) - Y_i(T_i = 0, \mathcal{G} = \emptyset))],$$

where one unit's utility gain is the difference between response under treatment with network interference and response under control without any network effects. Hence, learning an optimal policy that can not only make decisions on units but also adjust its decisions based on neighboring units is a challenging task.

The situation becomes more cumbersome when the policy needs to satisfy a specific capacity constraint. As another significant theoretical contribution, in [71], we establish guarantees for the regret of learned policies, both with and without treatment capacity

constraint. The main techniques applied for deriving the policy regret are concentration inequalities of networked random variables [51]. Using synthetic datasets with known data generation mechanisms, we show that policies learned from the population-averaged utility function that uses GNN-based causal estimators are more reliable and robust. The main reason is that the intervention policy for treatment assignment on a network might become very sensitive to the prediction accuracy of the employed causal estimators due to the interference effects.

Chapter 2

Embedding Models for Episodic Knowledge Graphs

Embedding Models for Episodic Knowledge Graphs

Yunpu Ma^{a,b}, Volker Tresp^{a,b}, Erik A. Daxberger^{1c}

^a*Siemens AG, Corporate Technology, Munich, Germany*

^b*Ludwig Maximilian University of Munich, Munich, Germany*

^c*ETH Zurich*

Abstract

In recent years a number of large-scale triple-oriented knowledge graphs have been generated and various models have been proposed to perform learning in those graphs. Most knowledge graphs are static and reflect the world in its current state. In reality, of course, the state of the world is changing: a healthy person becomes diagnosed with a disease and a new president is inaugurated. In this paper, we extend models for static knowledge graphs to temporal knowledge graphs. This enables us to store episodic data and to generalize to new facts (inductive learning). We generalize leading learning models for static knowledge graphs (i.e., Tucker, RESCAL, HolE, ComplEx, DistMult) to temporal knowledge graphs. In particular, we introduce a new tensor model, ConT, with superior generalization performance. The performances of all proposed models are analyzed on two different datasets: the Global Database of Events, Language, and Tone (GDELT) and the database for Integrated Conflict Early Warning System (ICEWS). We argue that temporal knowledge graph embeddings might be models also for cognitive episodic memory (facts we remember and can recollect) and that a semantic memory (*current* facts we know) can be generated from episodic memory by a marginalization operation. We validate this episodic-to-semantic projection hypothesis with the ICEWS dataset.

Keywords: knowledge graph, temporal knowledge graph, semantic memory, episodic memory, tensor models

¹Work done while at Siemens AG.

1. Introduction

In recent years a number of sizable Knowledge Graphs (KGs) have been developed, the largest ones containing more than 100 billion facts. Well known examples are DBpedia [1], YAGO [2], Freebase [3], Wikidata [4] and the Google KG [5]. Practical issues with completeness, quality and maintenance have been solved to a degree that some of these knowledge graphs support search, text understanding and question answering in large-scale commercial systems [5]. In addition, statistical embedding models have been developed that can be used to compress a knowledge graph, to derive implicit facts, to detect errors, and to support the above mentioned applications. A recent survey on KG models can be found in [6].

Most knowledge graphs are static and reflect the world at its current state. In reality, of course, the state of the world is changing: a healthy person becomes diagnosed with a disease and a new president is inaugurated. In this paper, we extend semantic knowledge graph embedding models to episodic/temporal knowledge graphs as an efficient way to store episodic data and to be able to generalize to new facts (inductive learning). In particular, we generalize leading approaches for static knowledge graphs (i.e., constrained Tucker, DistMult, RESCAL, HolE, ComplEx) to temporal knowledge graphs. We test these models using two temporal KGs. The first one is derived from the Integrated Conflict Early Warning System (ICEWS) data set which describes interactions between nations over several years. The second one is derived from the Global Database of Events, Language and Tone (GDELT) that, for more than 30 years, monitors news media from all over the world. In the experiments, we analyze the generalization abilities to new facts that might be missing in the temporal KGs and also analyze to what degree a factorized KG can serve as an explicit memory.

We propose that our technical models might be related to the brain’s explicit memory systems, i.e., its episodic and its semantic memory. Both are considered long-term memories and store information potentially over the life-time of an individual [7, 8, 9, 7]. The semantic memory stores general factual knowledge,

i.e., information we *know*, independent of the context where this knowledge was acquired and would be related to a static KG. Episodic memory concerns information we *remember* and includes the spatiotemporal context of events [10] and would correspond to a temporal KG.

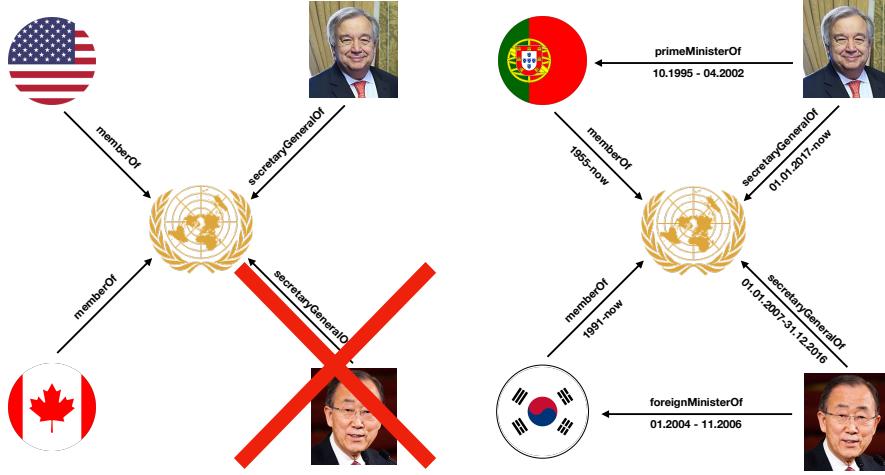


Figure 1: Illustrations of (left) a semantic knowledge graph and (right) an episodic knowledge graph. (Left) Every arrow represents a (subject, predicate, object) triple, with the annotation of the arrow denoting the respective predicate. The triple (Ban Ki-moon, SecretaryOf, UN) is deleted, since the knowledge graph has been updated with the triple (António Guterres, SecretaryOf, UN). (Right) Every arrow represents a (subject, predicate, object, timestamp) quadruple, where the arrow is both annotated with the respective predicate and timestamp. Here the quadruple involving is not deleted, since the attached timestamp reveals that the relationship is not valid at present.

An interesting question is how episodic and semantic memories are related. There is evidence that these main cognitive categories are partially dissociated from one another in the brain, as expressed in their differential sensitivity to brain damage. However, there is also evidence indicating that the different memory functions are not mutually independent and support one another [11]. We propose that semantic memory can be derived from episodic memory by marginalization. Hereby we also consider that many episodes describe starting and endpoints of state changes. For example, an individual might become sick

with a disease, which eventually is cured. Similarly, a president’s tenure eventually ends. We study our hypothesis on the Integrated Conflict Early Warning System (ICEWS) dataset, which contains many events with start and end dates. Figure 1 compares semantic and episodic knowledge graphs. Furthermore, Figure 2 illustrates the main ideas of building and modeling semantic and episodic knowledge graphs.

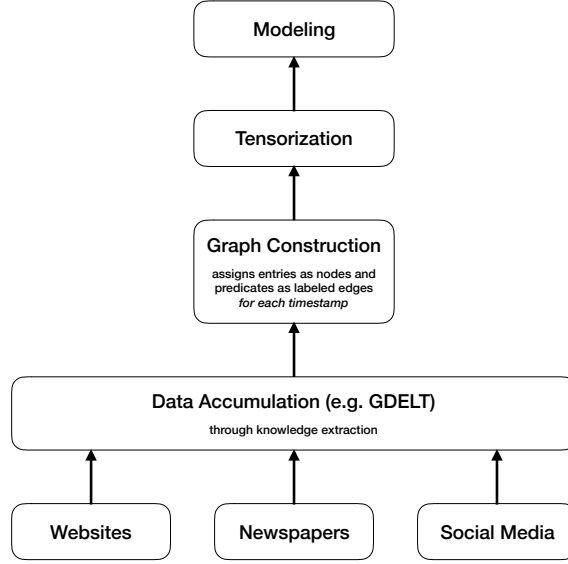


Figure 2: Illustration of the main idea behind the models presented in this paper. **Step 1:** Knowledge is extracted from unstructured data, such as websites, newspapers or social media. **Step 2:** The knowledge graph is constructed, where entities are assigned as nodes, and predicates as labeled edges; note that there is a labeled edge for each timestamp. **Step 3:** The knowledge graph is represented as a tensor; for semantic KGs, we obtain a 3-way tensor, storing (subject, predicate, object) triples, and for episodic KGs, we obtain a 4-way tensor, storing (subject, predicate, object, timestamp) quadruples. **Step 4:** The semantic and episodic tensors are decomposed and modeled via compositional or tensor models (see Section 2).

The paper is organized as follows. Section 2 introduces knowledge graphs,

the mapping of a knowledge graph to an adjacency tensor, and the statistical embedding models for knowledge graphs. We also describe how popular embedding models for KGs can be extended to episodic KGs. Section 3 shows experimental results on modelling episodic KGs. Finally, we present experiments on the possible relationships between episodic and semantic memory in Section 4.

2. Model Descriptions

A static or semantic knowledge graph (KG) is a triple-oriented knowledge representation. Here we consider a slight extension to the subject-predicate-object triple form by adding the value in the form $(e_s, e_p, e_o; \text{Value})$, where *Value* is a function of e_s, e_p, e_o and, e.g., can be a Boolean variable (*True* for 1, *False* for 0) or a real number. Thus $(\textit{Jack}, \textit{likes}, \textit{Mary}; \textit{True})$ states that Jack (the subject or head entity) likes Mary (the object or tail entity). Note that e_s and e_o represent the entities for subject index s and object index o . To simplify notation we also consider e_p to be a generalized entity associated with predicate type with index p . For the episodic KGs we introduce e_t , which is a generalized entity for time t .

To model a static KG, we introduce the three-way semantic adjacency tensor χ where the tensor element $x_{s,p,o}$ is the associated *Value* of the triple (e_s, e_p, e_o) . One can also define a companion tensor Θ_χ with the same dimensions as χ and with entries $\theta_{s,p,o}$. Thus, the probabilistic model for the semantic tensor χ is defined as $P(x_{s,p,o}|\theta_{s,p,o}) = \sigma(\theta_{s,p,o})$, where $\sigma(x) = 1/(1 + \exp(-x))$. Similarly, the four-way temporal or episodic tensor \mathcal{E} has elements $x_{t,s,p,o}$ which are the associated values of the quadruples (e_t, e_s, e_p, e_o) , with $t = 1, \dots, T$. Therefore, the probabilistic model for episodic tensor is defined with the corresponding companion tensor $\Theta_\mathcal{E}$ as

$$P(x_{t,s,p,o}|\theta_{t,s,p,o}) = \sigma(\theta_{t,s,p,o}) . \quad (1)$$

We assume that each entity e has a unique latent representation \mathbf{a} . In particular, the embedding approach used for modeling semantic and episodic knowledge

graphs assumes that $\theta_{s,p,o}^{sem} = f^{sem}(\mathbf{a}_{e_s}, \mathbf{a}_{e_p}, \mathbf{a}_{e_o})$, and $\theta_{t,s,p,o}^{epi} = f^{epi}(\mathbf{a}_{e_t}, \mathbf{a}_{e_s}, \mathbf{a}_{e_p}, \mathbf{a}_{e_o})$, respectively. Here, the indicator function $f^{sem/epi}(\cdot)$ is a function to be learned.

Given a labeled dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^m$, latent representations and other parameters (denoted as \mathcal{P}) are learned by minimizing the regularized logistic loss

$$\min_{\mathcal{P}} \sum_{i=1}^m \log(1 + \exp(-y_i \theta_i^{sem/epi})) + \lambda \|\mathcal{P}\|_2^2. \quad (2)$$

In general, most KGs only contain positive triples; non-existing triples are normally used as negative examples sampled with local closed-world assumption. Alternatively, we can minimize a margin-based ranking loss over the dataset such as

$$\min_{\mathcal{P}} \sum_{i \in \mathcal{D}_+} \sum_{j \in \mathcal{D}_-} \max(0, \gamma + \sigma(\theta_j^{sem/epi}) - \sigma(\theta_i^{sem/epi})), \quad (3)$$

where γ is the margin parameter, and \mathcal{D}_+ and \mathcal{D}_- denote the set of positive and negative samples, respectively.

There are different ways for modeling the indicator function $f^{epi}(\cdot)$ or $f^{sem}(\cdot)$. In this paper, we will only investigate multilinear models derived from tensor decompositions and compositional operations. We now describe the models in detail. Graphical illustrations of the described models are shown in Figure 3.

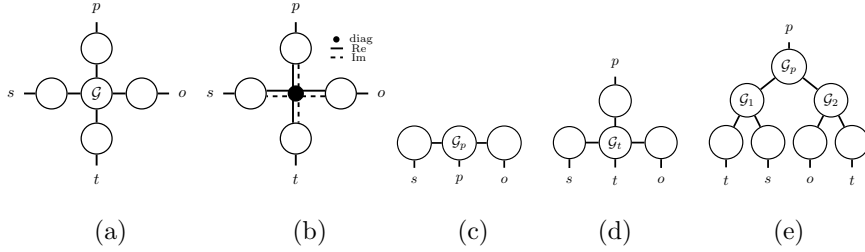


Figure 3: Illustrations of (a) episodic Tucker, (b) episodic ComplEx (where \bullet denotes contraction), (c) RESCAL, (d) ConT and (e) Tree. Each entity in the figure is represented as a circle with two edges, since the representation for an entity e is $\mathbf{a}_{e,i}$. In addition, \mathcal{G} represents the core tensor in Tucker, \mathcal{G}_p represents the matrix latent representation of predicate p in the RESCAL and Tree models, \mathcal{G}_t represents the three-dimensional tensor latent representation of timestamp t in the ConT model.

Table 1 and Table 2 summarize notations used throughout this paper for

easy reference, while Table 3 summarizes the number of parameters required for each model.²

Table 1: Summary of the general notations.

General	
Symbol	Meaning
e_s	Entity for subject index s
e_o	Entity for object index o
e_p	Generalized entity for predicate index p
e_t	Generalized entity for time index t
\mathbf{a}_{e_i}	Latent representation of entity e_i
$\mathbf{a}(e_{t_{start}})$	Latent representation of starting timestamp
a_{e_i, r_i}	r_i -th element of \mathbf{a}_{e_i}
\tilde{r}	Rank/Dimensionality of \mathbf{a}_{e_i} for $i \in \{s, p, o\}$
\tilde{r}_t	Rank/Dimensionality of \mathbf{a}_{e_t}
$N_{e/p/t}$	Number of entities / predicates / timestamps

Tucker. First, we consider the Tucker model for semantic tensor decomposition of the form $\theta_{s,p,o}^{sem} = \sum_{r_1, r_2, r_3=1}^{\tilde{r}} a_{e_s, r_1} a_{e_p, r_2} a_{e_o, r_3} g^{sem}(r_1, r_2, r_3)$. Here, $g^{sem}(r_1, r_2, r_3) \in \mathbb{R}$ are elements of the core tensor $\mathcal{G}^{sem} \in \mathbb{R}^{\tilde{r} \times \tilde{r} \times \tilde{r}}$. Similarly, the indicator function of a four-way Tucker model for episodic tensor decomposition is of the form

$$\theta_{t,s,p,o}^{epi} = \sum_{r_1=1}^{\tilde{r}_t} \sum_{r_2, r_3, r_4=1}^{\tilde{r}} a_{e_t, r_1} a_{e_s, r_2} a_{e_p, r_3} a_{e_o, r_4} g^{epi}(r_1, r_2, r_3, r_4), \quad (4)$$

with a four dimensional core tensor $\mathcal{G}^{epi} \in \mathbb{R}^{\tilde{r}_t \times \tilde{r} \times \tilde{r} \times \tilde{r}}$. Note that this is a con-

²For DistMult, ComplEx, and HolE it is required that $\tilde{r} = \tilde{r}_t$. In our experiments (see Sections 3 and 4), in order to enable a fair comparison between the different models, we assume that the latent representations of entities, predicates, and time indices all have the same rank/dimensionality.

Table 2: Summary of the notations for semantic and episodic knowledge graphs.

Semantic knowledge graphs		Episodic knowledge graphs	
Symbol	Meaning	Symbol	Meaning
χ	Sem. adjacency tensor	\mathcal{E}	Epi. adjacency tensor
Θ_χ	Companion tensor of χ	$\Theta_{\mathcal{E}}$	Companion tensor of \mathcal{E}
$x_{s,p,o}$	Value of (e_s, e_p, e_o)	$x_{t,s,p,o}$	Value of (e_t, e_s, e_p, e_o)
$\theta_{s,p,o}^{sem}$	Logit of (e_s, e_p, e_o)	$\theta_{t,s,p,o}^{epi}$	Logit of (e_t, e_s, e_p, e_o)
$f^{sem}(\cdot)$	Sem. indicator function	$f^{epi}(\cdot)$	Epi. indicator function
\mathcal{G}^{sem}	Sem. core tensor	\mathcal{G}^{epi}	Epi. core tensor
$g^{sem}(\cdot)$	Element of \mathcal{G}^{sem}	$g^{epi}(\cdot)$	Element of \mathcal{G}^{epi}

straint Tucker model, since, as in RESCAL, entities have unique representations, independent of the roles as subject or object.

RESCAL. Another model closely related to the semantic Tucker tensor decomposition is the RESCAL model, which has shown excellent performance in modelling KGs [12]. In RESCAL, subjects and objects have vector latent representations, while predicates have matrix latent representations. The indicator function of RESCAL for modeling semantic KGs takes the form $\theta_{s,p,o}^{sem} = \sum_{r_1, r_2=1}^{\tilde{r}} a_{e_s, r_1} g_p(r_1, r_2) a_{e_o, r_2}$, where $g_p(r_1, r_2)$ represents the matrix latent representation for the predicate e_p . Then next two models, Tree and ConT, are novel generalizations of RESCAL to episodic tensors.

Tree. From a practical perspective, training an episodic Tucker tensor model is very expensive since the computational complexity is approximately \tilde{r}^4 . Tensor networks provide a general and flexible framework to design nonstandard tensor decompositions [13, 14]. One of the simplest tensor networks is a tree tensor decomposition (\mathcal{T}) of the episodic indicator function, which is illustrated in compositional operations. We now describe the models in detail. Graphical illustrations of the described models are shown in Figure 3(e). Therefore, we propose a tree tensor decomposition (\mathcal{T}) of the episodic indicator function. The tree \mathcal{T} is partitioned into two subtrees \mathcal{T}_1 and \mathcal{T}_2 , wherein subject e_s and time

e_t reside in \mathcal{T}_1 , while object e_o and an auxiliary time e_t reside in \mathcal{T}_2 . \mathcal{T}_1 and \mathcal{T}_2 are connected with e_p through two core tensors \mathcal{G}_1 and \mathcal{G}_2 . Thus, the indicator function can be written as

$$\theta_{t,s,p,o}^{epi} = \sum_{r_1, r_6=1}^{\tilde{r}_t} \sum_{r_2, r_3, r_4, r_5=1}^{\tilde{r}} a_{e_t, r_1} a_{e_s, r_2} g_1(r_1, r_2, r_3) g_p(r_3, r_4) g_2(r_4, r_5, r_6) a_{e_o, r_5} a_{e_t, r_6}. \quad (5)$$

Within \mathcal{T} , we reduce the four-way core tensor in Tucker into two three-dimensional tensors \mathcal{G}_1 and \mathcal{G}_2 , so that the computational complexity of \mathcal{T} is approximately \tilde{r}^3 .

ConT. ConT is another generalization of the RESCAL model to episodic tensors with reduced computational complexity of approximately \tilde{r}^3 . The idea is that another way of reducing the complexity is by contracting indices of the core tensor. Therefore, we contract the \mathcal{G} from Tucker with the time index giving a three-way core tensor \mathcal{G}_t for each time instance. The indicator function takes the form

$$\theta_{t,s,p,o}^{epi} = \sum_{r_1, r_2, r_3=1}^{\tilde{r}} a_{e_s, r_1} a_{e_p, r_2} a_{e_o, r_3} g_t(r_1, r_2, r_3). \quad (6)$$

In this model, the tensor \mathcal{G}_t resembles the relation-specific matrix \mathcal{G}_p from RESCAL. Later, we will see that ConT is a superior model for modeling episodic knowledge graphs due to the representational flexibility of its high-dimensional tensor \mathcal{G}_t for the time index.

Even though the complexity of Tree and ConT is reduced as compared to episodic Tucker, the three-dimensional core tensor might cause rapid overfitting during training. Therefore, we next propose episodic generalization of compositional models, such as DistMult [15], HolE [16] and ComplEx [17]. For those models, the number of parameters only increases linearly with the rank.

DistMult. DistMult [15] is a simple generalization of the CP model, by enforcing the constraint that entities should have unique representations. Episodic DistMult takes the form $\theta_{t,s,p,o}^{epi} = \sum_{i=1}^{\tilde{r}} \lambda_i a_{e_t, i} a_{e_s, i} a_{e_p, i} a_{e_o, i}$. Here, we require that vector latent representations of entities, predicates, and timestamps have

the same rank. DistMult is a special case of Tucker having a core tensor with only diagonal elements λ_i .

HolE. Holographic embedding (HolE) [16] is a state-of-art link prediction and knowledge graph completion method, which is inspired by holographic models of associative memory.

HolE uses circular correlation to generate a compositional representation from inputs e_s and e_o . The indicator of HolE reads $\theta_{s,p,o}^{sem} = \mathbf{a}_{e_p} \cdot (\mathbf{a}_{e_s} \star \mathbf{a}_{e_o})$, where $\star : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}^d$ denotes the circular correlation $[\mathbf{a} \star \mathbf{b}]_k = \sum_{i=0}^{d-1} a_i b_{(k+i) \bmod d}$. We define the episodic extension of HolE as

$$\theta_{t,s,p,o}^{epi} = \mathbf{a}_{e_t} \cdot (\mathbf{a}_{e_p} \star (\mathbf{a}_{e_s} \star \mathbf{a}_{e_o})). \quad (7)$$

As argued by [16], HolE employs a holographic reduced representation [18] to store and retrieve the predicates from e_s and e_o . Analogously, episodic HolE should be able to retrieve the stored timestamps from e_p , e_s and e_o . In the semantic case, e_p can be retrieved if existing triple relations are stored via circular convolution \star , and superposition in the representation $\mathbf{a}_{e_o} = \sum_{(s,p) \in \mathcal{S}_o} \mathbf{a}_{e_p} \star \mathbf{a}_{e_s}$, where \mathcal{S}_o is the set of all true triples given e_o . This is based on the fact that $\mathbf{a} \star \mathbf{a} \approx \delta$ [16]. Analogously, the stored timestamp e_t for an event can be retrieved if all existing episodic events are stored via \star , and superposition in the representation of e_o , $\mathbf{a}_{e_o} = \sum_{(t,s,p) \in \mathcal{S}_o} \mathbf{a}_{e_t} \star (\mathbf{a}_{e_p} \star \mathbf{a}_{e_s})$, where \mathcal{S}_o is the set of all true quadruples (t, s, p, o) given e_o . However, high order circular correlation/convolution will increase the inaccuracy of retrieval. Another motivation for our episodic extension (7) is that a compositional operator of the form $\mathbf{a}_{e_t} \cdot \tilde{f}$ allows a projection from episodic memory to semantic memory, to be detailed later.

ComplEx. Complex embedding (ComplEx) [17] is another state-of-art method closely related to HolE. It can accurately describe both symmetric and antisymmetric relations. HolE is a special case of ComplEx with imposed conjugate symmetry on embeddings [19]. Thus, ComplEx has more degrees of freedom, if compared to HolE. For the semantic complex embedding, the indicator function is $\theta_{s,p,o}^{sem} = \text{Re} \left(\sum_i^{\tilde{r}} a_{e_s,i} a_{e_p,i} \bar{a}_{e_o,i} \right)$ with complex valued \mathbf{a} and where

the bar indicates the complex conjugate. To be consistent with the episodic HolE, the episodic complex embedding is defined as³

$$\theta_{t,s,p,o}^{epi} = \text{Re} \left(\sum_i^{\tilde{r}} a_{e_t,i} a_{e_s,i} a_{e_p,i} \bar{a}_{e_o,i} \right). \quad (8)$$

3. Experiments on Episodic Models

We investigate the proposed tensor and compositional models with experiments which are evaluated on two datasets:

ICEWS. The Integrated Conflict Early Warning System (ICEWS) dataset [20] is a natural episodic dataset recording dyadic events between different countries. An example entry could be (*Turkey, Syria, Fight, 12/25/2014*). These dyadic events are aggregated into a four-way tensor \mathcal{E} with 258 entities, 20 relation types, and 72 timestamps, which has in total 320,118 positive (e_t, e_s, e_p, e_o) quadruples⁴. This dataset was first created and used in [21]. From this ICEWS dataset, a semantic tensor is generated by extracting consecutive events that last until the last timestamp, constituting the *current*⁵ semantic facts of the world.

GDELT. The Global Database of Events, Language and Tone (GDELT) [20] monitors the world’s news media in broadcast, print and web formats from all over the world, daily since January 1, 1979⁶. We use GDELT as a large episodic dataset. For our experiments, GDELT data is collected from January 1, 2012 to December 31, 2012 (with a temporal granularity of 24 hrs). These events are aggregated into an episodic tensor \mathcal{E} with 1100 entities, 180 relation

³One can show that Eq. (7) is equivalent to Eq. (8) by converting it to the frequency domain [19]. Then, $\theta_{t,s,p,o}^{epi} \propto \omega_{e_t}^T (\bar{\omega}_{e_p} \odot \bar{\omega}_{e_s} \odot \omega_{e_o})$, where $\omega = \mathcal{F}(\mathbf{a}) \in \mathbb{C}^{\tilde{r}}$ are the discrete Fourier transforms of embeddings \mathbf{a} , and using the fact that ω is conjugate symmetric for real vector \mathbf{a} .

⁴Note that for an episodic event the dataset contains all the quadruples (e_{t_i}, e_s, e_p, e_o) for $t_i \in \{t_{start}, t_{start} + 1, \dots, t_{end} - 1, t_{end}\}$.

⁵*Current* always indicates the last timestamp/timestamps of the applied episodic KGs.

⁶<https://www.gdeltproject.org/about.html>

Table 3: Number of parameters for different models and the runtime of one training epoch on the GDELT dataset.

Model	Semantic	Episodic	Complexity	Runtime		
				rank 40	rank 60	rank 150
DistMult	$(N_e + N_p + 1)\tilde{r}$	$(N_e + N_p + N_t + 1)\tilde{r}$	$\mathcal{O}(\tilde{r})$	35.2s	36.4s	53.7s
HolE	$(N_e + N_p)\tilde{r}$	$(N_e + N_p)\tilde{r}$	$\mathcal{O}(\tilde{r} \log \tilde{r})$	42.8s	43.2s	59.0s
ComplEx	$2(N_e + N_p)\tilde{r}$	$2(N_e + N_p + N_t)\tilde{r}$	$\mathcal{O}(\tilde{r})$	40.1s	42.4s	57.5s
Tree	—	$N_e\tilde{r} + N_p\tilde{r}^2 + (N_t + 2\tilde{r}^2)\tilde{r}_t$	$\mathcal{O}(\tilde{r}^3)$	133.6s	160.2s	—
ConT	—	$(N_e + N_p)\tilde{r} + N_t\tilde{r}^3$	$\mathcal{O}(\tilde{r}^3)$	95.4s	226.1s	—
Tucker	$(N_e + N_p)\tilde{r} + \tilde{r}^3$	$(N_e + N_p)\tilde{r} + (N_t + \tilde{r}^3)\tilde{r}_t$	$\mathcal{O}(\tilde{r}^4)$	144.2s	387.9s	—

types, and 366 timestamps, which has in total 2,563,561 positive (e_t, e_s, e_p, e_o) quadruples.

We assess the quality of episodic information retrieval on both datasets for the proposed tensor and compositional models. Since both episodic datasets only consist of positive quadruples, we generated negative episodic instances following the protocol of corrupting semantic triples given by Bordes [22]: negative instances of an episodic quadruple (e_s, e_p, e_o, e_t) are drawn by corrupting the object e_o to $e_{o'}$ or the timestamp e_t to $e_{t'}$, meaning that $(e_s, e_p, e_{o'}, e_t)$ serves as a negative evidence of the episodic event at time instance e_t , and $(e_s, e_p, e_o, e_{t'})$ is a true fact which cannot be correctly recalled at time instance $e_{t'}$. During training, for each positive sample in a batch we assigned two negative samples with corrupted object or corrupted subject.

The model performance is evaluated using the following scores. To retrieve the occurrence time, for each true quadruple, we replace the time index e_t with every other possible time index $e_{t'}$, compute the value of the indicator function $\theta_{t',s,p,o}^{epi}$ and rank them in a decreasing order. We filter the ranking as in [22] by removing all quadruples where $x_{t',s,p,o} = 1$ and $t \neq t'$, in order to eliminate ambiguity during episodic information retrieval. Similarly, we evaluated the retrieval of the predicate between a given subject and object at a certain time instance by computing and ranking the indicator $\theta_{t,s,p',o}^{epi}$. We also evaluated the

retrieval of entities by ranking and averaging the filtered indicators $\theta_{t,s',p,o}$ and $\theta_{t,s,p,o'}$. To measure the generalization ability of the models, we report different measures of the ranking: mean reciprocal rank (MRR), and Hits@n on the test dataset.

The datasets were split into train, validation, and test sets that contain the most frequently appearing entities in the episodic knowledge graphs. Training was performed by minimizing the logistic loss (2), and was terminated using early stopping on the validation dataset by monitoring the filtered MRR recall scores every $\{50, 100\}$ epochs depending on the models, where the maximum training duration was 500 epochs. This ensures that the generalization ability of unique latent representations of entities doesn't suffer from overfitting. Before training, all model parameters are initialized using Xavier initialization [23]. We also apply an l_2 norm penalty on all parameters for regularization purposes (see Eq. (2)).

In Table 3 we summarize the runtime for one training epoch on the GDEL T dataset for different models at ranks $\tilde{r} = \tilde{r}_t \in \{40, 60, 150\}$. All experiments were performed on a single Tesla K80 GPU. In the following experiments, for compositional models we search rank in $\{100, 150\}$, while for tensor models we search optimal rank in $\{40, 50, 60\}$ since larger ranks could lead to overfitting rapidly. Loss function is minimized with Adam method [24] with the learning rate selected from $\{0.001, 1e - 4, 5e - 5\}$.

We first assess the filtered MRR, Hits@1, Hits@3, and Hits@10 scores of inferring missing entities and predicates on the GDEL T test dataset. Table 4 summarizes the results. Generalizations on the test dataset indicate the inductive reasoning capability of the proposed models. This generalization can be useful for the completion of evolving KGs with missing records, such as clinical datasets. It can be seen that tensor models are able to outperform compositional models consistently on both entity and predicate prediction tasks. ConT has the best inference results on the entity-related tasks, while Tucker performs better on the predicate-related tasks. The superior Hits@1 result of ConT on the entity prediction indicates that there are easily to be fitted entities in the GDEL T

Table 4: Filtered results of inferring missing entities and predicates of episodic quadruples evaluated on the GDELT dataset.

Method	Entity				Predicate			
	MRR	@1	@3	@10	MRR	@1	@3	@10
DistMult	0.182	6.55	19.77	43.70	0.269	12.65	30.29	59.40
HolE	0.177	6.67	18.95	41.84	0.256	11.81	28.35	57.73
ComplEx	0.172	6.54	17.52	41.56	0.255	12.05	27.75	56.60
Tree	0.196	8.17	21.00	44.65	0.274	13.30	30.66	60.05
Tucker	0.204	8.93	21.85	46.35	0.275	12.69	31.35	60.70
ConT	0.233	13.85	24.65	42.96	0.263	12.83	29.27	57.30

Table 5: Filtered results for entities and predicates recollection/prediction evaluated on the ICEWS dataset.

Method	Entity				Predicate			
	MRR	@1	@3	@10	MRR	@1	@3	@10
DistMult	0.222	9.72	22.48	52.32	0.520	33.73	62.25	91.13
HolE	0.229	9.85	23.49	54.21	0.517	31.55	65.47	93.59
ComplEx	0.229	8.94	23.53	57.72	0.506	30.99	61.46	93.44
Tree	0.205	10.48	19.84	42.81	0.554	36.62	67.25	94.70
Tucker	0.257	12.88	27.10	54.43	0.563	36.96	69.55	95.43
ConT	0.264	15.71	29.60	46.67	0.557	38.12	67.76	87.71

dataset along the timestamps. In fact, the GDELT dataset is unbalanced, and episodic quadruples related to certain entities dominate in the episodic Knowledge graph, such as quadruples containing the entities *USA*, or *UN*. Experiment results on balanced and extremely sparse episodic dataset will be reported in the following.

Next, Table 5 shows the MRR, Hits@1, Hits@3, and Hits@10 scores of inferring missing entities and predicates on the ICEWS test dataset. Similarly, we can read that tensor models outperform compositional models on both missing entity and predicate inference tasks. The superior Hits@1 result of ConT for the

missing entity prediction indicates again that the ICEWS dataset is unbalanced, and episodic quadruples related to certain entities dominate.

Table 6: Filtered recall scores for entities and timestamps recollection on the ICEWS (rare) training dataset.

Method	Rank	Timestamp		Entity	
		MRR	@3	MRR	@3
DistMult	200	0.257	27.0	0.211	21.9
HolE	200	0.216	20.8	0.179	16.3
ComplEx	200	0.354	40.3	0.301	33.2
Tree	40	0.421	55.3	0.314	35.7
Tucker	40	0.923	98.9	0.893	97.1
ConT	40	0.982	99.7	0.950	97.9

The recollection of the exact occurrence time of a significant past event (e.g. unusual, novel, attached with emotion) is also an important capability of episodic cognitive memory function. In order to manifest this perspective of proposed models, Table 6 shows the filtered MRR, and Hits@3 scores for the timestamps and entities recollection on the episodic ICEWS (rare) training dataset, where rank column registers the optimal and minimum rank $\tilde{r} = \tilde{r}_t$ having the outstanding recall scores. Figure 4 further displays the filtered MRR score as a function of rank. Unlike the original ICEWS, which contains many consecutive events that last from the first to the last timestamp leading to unreasonably high filtered timestamp recall scores, this ICEWS (rare) dataset consists of rare temporal events that happen less than three times throughout the whole time and starting points of events.

The outstanding performance of ConT compared with other compositional models indicates the importance of large dimensionality of time latent representation for the episodic tensor reconstruction / episodic memory recollection. Recall that for ConT the real *dimension* of the latent representation of time is actually \tilde{r}^3 after flattening \mathcal{G}_t . This flexible latent representation for time could

compress almost all the semantic triples that occur at a certain instance ⁷.

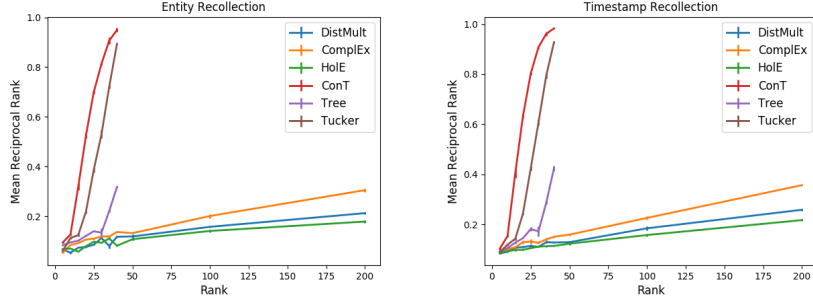


Figure 4: Filtered MRR scores vs. rank for the entities (left) and timestamps (right) recollection on the ICEWS (rare) training dataset.

4. Semantic Memory from Episodic Memory with Marginalization

We already discussed that a semantic KG might be related to a human semantic memory and that an episodic KG might be related to a human episodic memory. It has been speculated that episodic and semantic memory must be closely related, and that semantic memory is generated from episodic memory by some training process [28, 29]. As a very simple implementation of that idea, we propose that a semantic memory could be generated from episodic memory by marginalizing time. Thus, both types of memories would rely on identical representations and the marginalization step can be easily performed: Since probabilistic tensor models belong to the classes of sum-product nets, a marginalization simply means an integration over all time representations.

Thus, in the second set of experiments, we test the hypothesis that semantic

⁷This observation has its biological counterpart. In fact, the entorhinal cortex, which plays an important role in the formation of episodic memory, is the main part of the adult hippocampus that shows neurogenesis [25]. In an adult human, approximately 700 new neurons are added per day through hippocampal neurogenesis, which are believed to perform sensory and spatial information encoding, as well as temporal separation of events [26, 27].

memory can be derived from episodic memory by projection. In other words, a semantic knowledge graph containing *current* semantic facts can be approximately constructed after modeling a corresponding episodic knowledge graph via marginalization. A marginalization can be performed by activating all time index neurons, i.e., summing over all \mathbf{a}_{e_t} , since, e.g., Tucker decompositions are an instance of a so-called sum-product network [30]. However, events having start as well as end timestamps cannot simply be integrated into our *current* semantic knowledge describing what we *know* now. For example, (Ban Ki-moon, SecretaryOf, UN) is not consistent with what we *know* currently. To resolve this problem, we introduce two types of time indices, $e_{t_{start}}$ and $e_{t_{end}}$, having the latent representations $\mathbf{a}(e_{t_{start}})$ and $\mathbf{a}(e_{t_{end}})$, respectively. Those time indices can be used to construct the episodic tensor \mathcal{E}_{start} aggregating the start timestamps of consecutive events, as well as the episodic tensor \mathcal{E}_{end} aggregating the end timestamps⁸.

For the projection, instead of only summing over $\mathbf{a}(e_{t_{start}})$, we also subtract the sum over $\mathbf{a}(e_{t_{end}})$. In this way, we can achieve the effect that events that have terminated already (i.e., have an end time index smaller than the current time index) are not integrated into the current semantic facts. Now, to test our hypothesis that this extended projection allows us to derive semantic memory from episodic memory, we trained HolE, DistMult, ComplEx, ConT, and Tucker on the episodic tensors \mathcal{E}_{start} and \mathcal{E}_{end} as well as on the semantic tensor χ derived from ICEWS. Note that only these models allow projection, since their indicator functions can be written in the form $\theta_{t,s,p,o}^{epi} = \mathbf{a}_{e_t} \cdot \tilde{f}$, where \tilde{f} can be arbitrary function of \mathbf{a}_{e_s} , \mathbf{a}_{e_p} , and \mathbf{a}_{e_o} depending on the model choice⁹. The

⁸E.g., if the duration of a triple event (e_s, e_p, e_o) lasts from t_{start} to t_{end} , the quadruple $(e_s, e_p, e_o, e_{t_{start}})$ is stored in \mathcal{E}_{start} , while $(e_s, e_p, e_o, e_{t_{end}})$ is stored in \mathcal{E}_{end} only if $t_{end} < T$ (where T is the last timestamp). In other words, events that last until the last timestamp do not possess e_{end} .

⁹For ConT, $\theta_{t,s,p,o}^{epi} = \text{flatten}(g_t) \cdot (\mathbf{a}_{e_s} \otimes \mathbf{a}_{e_p} \otimes \mathbf{a}_{e_o})$, where \otimes denotes the outer product. For ComplEx, $\theta_{t,s,p,o}^{epi} = \text{Re}(\mathbf{a}_{e_t}) \cdot \text{Re}(\mathbf{a}_{e_s} \odot \mathbf{a}_{e_p} \odot \bar{\mathbf{a}}_{e_o}) - \text{Im}(\mathbf{a}_{e_t}) \cdot \text{Im}(\mathbf{a}_{e_s} \odot \mathbf{a}_{e_p} \odot \bar{\mathbf{a}}_{e_o})$, where \odot denotes the Hadamard product. The Tree model cannot be written in this form since e_t

model parameters are optimized using the margin-based ranking loss (3)¹⁰.

Training was first performed on the episodic tensor \mathcal{E}_{start} , and then on \mathcal{E}_{end} with *fixed* \mathbf{a}_{e_s} , \mathbf{a}_{e_p} , and \mathbf{a}_{e_o} obtained from the training on \mathcal{E}_{start} , since we assume that latent representations for subject, object, and predicate of a consecutive event do not change during the event. Note that after training in this way, we could recall the starting and terminal point of a consecutive event (see the episodic tensor reconstruction experiments in Section 3), or infer a *current* semantic fact solely from the latent representations instead of rule-based reasoning.

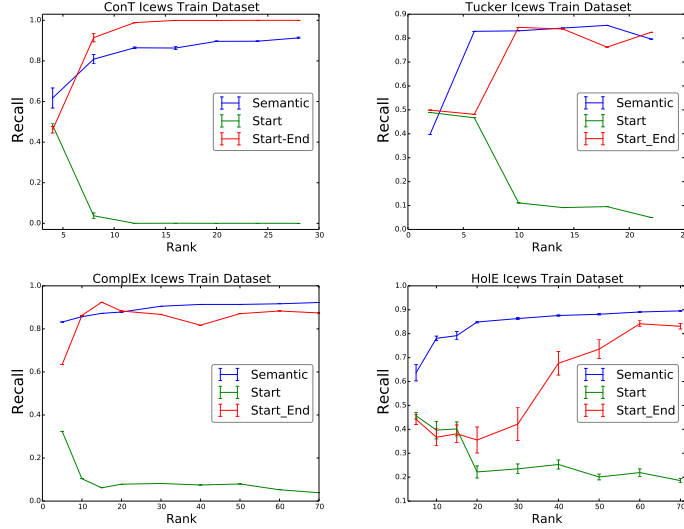


Figure 5: Recall scores vs. rank for the episodic-to-semantic projection on the ICEWS dataset with two different projection methods.

To evaluate the projection, we compute the recall and area under precision-recall-curve (AUPRC) scores for the projection at different ranks on the ICEWS

resides in both subtrees \mathcal{T}_1 and \mathcal{T}_2 .

¹⁰For the projection experiment, we omit the sigmoid function in Eq. (3), train and interpret the multilinear indicator $\theta_{t,s,p,o}^{epi} = \mathbf{a}_{e_t} \cdot \tilde{f}(\mathbf{a}_{e_s}, \mathbf{a}_{e_p}, \mathbf{a}_{e_o})$ directly as the probability of episodic quadruple. Only in this way of training, a projection is mathematically legitimate.

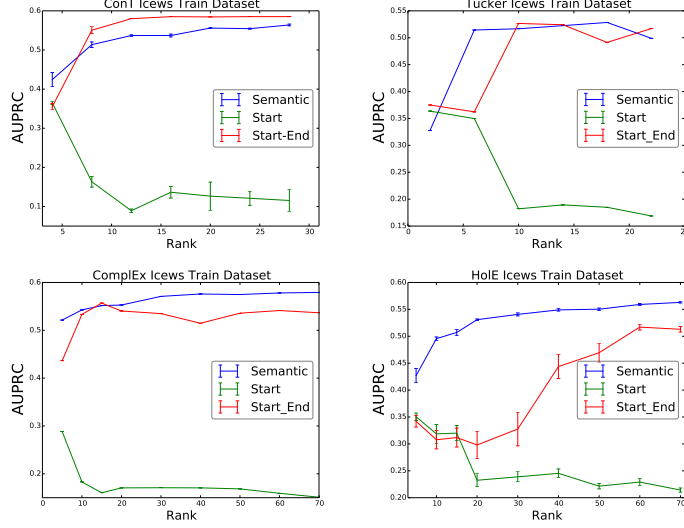


Figure 6: AUPRC scores vs. rank for the episodic-to-semantic projection on the ICEWS dataset with two different projection methods.

training dataset, and compare them with the scores obtained from training the semantic tensor separately. The semantic dataset contains positive triples, which are episodic events that continue until the last (current) timestamp, e.g. (António Guterres, SecretaryOf, UN, *True*), along with negative triples extracted from already terminated episodic events, e.g. (Ban Ki-moon, SecretaryOf, UN, *False*). During the test phase of projection, a triple from the semantic dataset is given with non-specified time index, e.g. $(e_s, e_p, e_o, \text{True/False}, t)$. Then, for the first method considering only the starting point of an episodic event, the projection to semantic space is computed as

$$\theta_{s,p,o}^{proj} = [\sum_{t_{start}=1}^T \mathbf{a}(e_{t_{start}})] \cdot \tilde{f}, \quad (9)$$

while for the second method considering both starting and terminal points, the projection is computed as

$$\theta_{s,p,o}^{proj} = \left[\sum_{t_{start}=1}^T \mathbf{a}(e_{t_{start}}) - \sum_{t_{end}=1}^T \mathbf{a}(e_{t_{end}}) \right] \cdot \tilde{f}. \quad (10)$$

Then, the scores are evaluated by taking the label of the given semantic triple as the target, and taking $\theta_{s,p,o}^{proj}$ as the prediction. The goal of this test is to check how well the algorithms can project a given consecutive event $(e_s, e_p, e_o, t_{start} \cdots t_{end})$ to semantic knowledge space using only the marginalized latent representation of time. All other experimental settings are similar to those in Section 3, and the experiments were repeated four times on different sampled training datasets.

Figure 5 shows the recall scores for the two different projection methods on the training dataset in comparison to the separately trained semantic dataset. Due to limited space, we only show four models: ConT, Tucker, ComplEx, and HolE. As we can see, only the marginalization considering both starting and terminal time indices allows a reasonable projection from episodic memory to the *current* semantic memory. Again, ConT¹¹ exhibits the best performance, with its recall score saturating after $\tilde{r} \approx 15$. In contrast, HolE shows insufficient projection quality with sizable errors, especially at small ranks, which is due to its higher-order encoding noise. To show that the two types of latent representations of time do not simply eliminate each other for a correct episodic projection, Figure 6 shows the AUPRC scores evaluated on the training dataset. Overall, this experiment supports the idea that semantic memory is a long-term storage for episodic memory, where the exact timing information is lost.

For a fair comparison, in the last experiment we report the recall scores of the semantic models obtained by projecting the episodic models with respect to the temporal dimension. We compare two projection methods, the Start projection which only considers the starting point of episodic events (see Eq. 9), and the Start-End projection which takes both the starting and terminal points of episodic events into consideration. In addition, we report the recall scores on two semantic datasets. The first one contains genuine semantic facts, while the second dataset contains false semantic triples which should already be ruled out

¹¹Note that since ConT doesn't have a direct semantic counterpart, we instead use the semantic results obtained using RESCAL. This is reasonable since ConT can be viewed as a high-dimensional (i.e., episodic) generalization of RESCAL.

Table 7: Filtered and raw Hits@10 scores for the episodic-to-semantic projection. Two projection methods, Start (Eq. 9), Start-End (Eq. 10), are compared. Furthermore, semantic ICEWS dataset with genuine semantic triples, and semantic ICEWS dataset with false triples are used for the projection experiments. Various projection scores are compared with the scores which are obtained by directly modeling the semantic ICEWS dataset with genuine semantic triples.

Method	Start		Start-End		Start (false)		Start-End (false)		Semantic	
	Filter	Raw	Filter	Raw	Filter	Raw	Filter	Raw	Filter	Raw
DistMult	3.8	3.6	5.6	5.0	4.0	3.8	3.8	3.6	59.3	32.4
HolE	5.8	5.4	5.5	5.1	4.7	4.5	5.6	5.2	56.1	31.3
ComplEx	4.1	3.7	4.9	4.4	3.9	3.7	3.8	3.6	60.1	29.4
Tucker	14.8	13.1	15.1	13.4	11.3	10.3	11.8	10.9	46.5	23.7
ConT	30.9	24.6	40.8	30.3	23.0	19.9	22.6	19.3	43.8	20.4

through the projection.

Two different projections are performed on two semantic datasets, the genuine one and the false one. Theoretically, the recall scores on the genuine semantic dataset should be higher than those on the false dataset. Thus, the model hyper-parameters are chosen by monitoring the difference between the recall scores Hits@10 on the genuine and false semantic datasets.

Table. 7 reports the filtered and raw Hits@10 metrics for different models, projection methods, and datasets. Moreover, we also compare the projection with the recall scores obtained by directly modeling the genuine semantic dataset using the corresponding semantic models ¹². The ConT model has the best projection performance, since its projected recall scores on the genuine dataset are much higher than those obtained on the false semantic dataset. Moreover, the Start-End projection method based on the ConT model is the only combination which achieves similar results compared to the corresponding semantic model. One can also notice that all the projected compositional models are only able to tell whether a semantic triple is already ruled out or not before the last

¹²Note that we use the RESCAL model as the corresponding semantic model for the ConT.

timestamp, however they can not provide good inference results on the genuine semantic dataset.

5. Conclusion

This paper described the first mathematical models for the declarative memories: the semantic and episodic memory functions. To model these cognitive functions, we generalized leading approaches for static knowledge graphs (i.e., Tucker, RESCAL, HolE, ComplEx, DistMult) to 4-dimensional temporal/episodic knowledge graphs. In addition, we developed two novel generalizations of RESCAL to episodic tensors, i.e., Tree and ConT. In particular, ConT has superior performance overall, which indicates the importance of introduced high-dimensional latent representation of time for both sparse episodic tensor reconstruction and generalization.

Our hypothesis is that perception includes an active semantic decoding process, which relies on latent representations of entities and predicates, and that episodic and semantic memories depend on the same decoding process. We argue that temporal knowledge graph embeddings might be models for human cognitive episodic memory and that semantic memory (facts we know) can be generated from episodic memory by a marginalization operation. We also test this hypothesis on the ICEWS dataset, the experiments show that the *current* semantic facts can only be derived from the episodic tensor by a proper projection considering both starting and terminal points of consecutive events.

Acknowledgements. This work is funded by the *Cognitive Deep Learning* research project in Siemens AG.

References

References

- [1] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, Z. Ives, Dbpedia: A nucleus for a web of open data, *The semantic web* (2007) 722–735.
- [2] F. M. Suchanek, G. Kasneci, G. Weikum, Yago: a core of semantic knowledge, in: *Proceedings of the 16th international conference on World Wide Web*, ACM, 2007, pp. 697–706.
- [3] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, J. Taylor, Freebase: a collaboratively created graph database for structuring human knowledge, in: *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, AcM, 2008, pp. 1247–1250.
- [4] D. Vrandečić, M. Krötzsch, Wikidata: a free collaborative knowledgebase, *Communications of the ACM* 57 (10) (2014) 78–85.
- [5] A. Singhal, Introducing the knowledge graph: things, not strings, *Official google blog*.
- [6] M. Nickel, K. Murphy, V. Tresp, E. Gabrilovich, A review of relational machine learning for knowledge graphs, *Proceedings of the IEEE*.
- [7] H. Ebbinghaus, *Über das gedächtnis: untersuchungen zur experimentellen psychologie*, Duncker & Humblot, 1885.
- [8] R. C. Atkinson, R. M. Shiffrin, Human memory: A proposed system and its control processes, *Psychology of learning and motivation* 2 (1968) 89–195.
- [9] L. R. Squire, *Memory and brain*.
- [10] E. Tulving, Episodic and semantic memory: Where should we go from here?, *Behavioral and Brain Sciences* 9 (03) (1986) 573–577.

- [11] D. L. Greenberg, M. Verfaellie, Interdependence of episodic and semantic memory: evidence from neuropsychology, *Journal of the International Neuropsychological society* 16 (05) (2010) 748–753.
- [12] M. Nickel, V. Tresp, H.-P. Kriegel, A three-way model for collective learning on multi-relational data, in: *Proceedings of the 28th international conference on machine learning (ICML-11)*, 2011, pp. 809–816.
- [13] A. Cichocki, Era of big data processing: A new approach via tensor networks and tensor decompositions, in: *International Workshop on Smart Info-Media Systems in Asia (SISA-2013)*, 2013.
- [14] A. Cichocki, Tensor networks for big data analytic and large-scale optimization problems, in: *Second Int. Conference on Engineering and Computational Schematics (ECM2013)*, 2013.
- [15] B. Yang, W.-t. Yih, X. He, J. Gao, L. Deng, Embedding entities and relations for learning and inference in knowledge bases, *International Conference on Learning Representations (ICLR)*.
- [16] M. Nickel, L. Rosasco, T. Poggio, Holographic embeddings of knowledge graphs, in: *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.
- [17] T. Trouillon, J. Welbl, S. Riedel, É. Gaussier, G. Bouchard, Complex embeddings for simple link prediction, in: *International Conference on Machine Learning*, 2016, pp. 2071–2080.
- [18] T. A. Plate, Holographic reduced representations, *IEEE Transactions on Neural Networks* 6 (3) (1995) 623–641.
- [19] K. Hayashi, M. Shimbo, On the equivalence of holographic and complex embeddings for link prediction, *CoRR* abs/1702.05563.
URL <http://arxiv.org/abs/1702.05563>
- [20] M. D. Ward, A. Beger, J. Cutler, M. Dickenson, C. Dorff, B. Radford, Comparing gdel and icews event data, *Analysis* 21 (2013) 267–297.

- [21] A. Schein, J. Paisley, D. M. Blei, H. Wallach, Bayesian poisson tensor factorization for inferring multilateral relations from sparse dyadic event counts, in: *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM, 2015, pp. 1045–1054.
- [22] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, O. Yakhnenko, Translating embeddings for modeling multi-relational data, in: *Advances in neural information processing systems*, 2013, pp. 2787–2795.
- [23] X. Glorot, Y. Bengio, Understanding the difficulty of training deep feed-forward neural networks., in: *Aistats*, Vol. 9, 2010, pp. 249–256.
- [24] D. Kingma, J. Ba, Adam: A method for stochastic optimization, *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*.
- [25] W. Deng, J. B. Aimone, F. H. Gage, New neurons and new memories: how does adult hippocampal neurogenesis affect learning and memory?, *Nature reviews. Neuroscience* 11 (5) (2010) 339.
- [26] O. Lazarov, C. Hollands, Hippocampal neurogenesis: learning to remember, *Progress in neurobiology* 138 (2016) 1–18.
- [27] K. L. Spalding, O. Bergmann, K. Alkass, S. Bernard, M. Salehpour, H. B. Huttner, E. Boström, I. Westerlund, C. Vial, B. A. Buchholz, et al., Dynamics of hippocampal neurogenesis in adult humans, *Cell* 153 (6) (2013) 1219–1227.
- [28] J. L. McClelland, B. L. McNaughton, R. C. O’reilly, Why there are complementary learning systems in the hippocampus and neocortex: insights from the successes and failures of connectionist models of learning and memory., *Psychological review* 102 (3) (1995) 419.
- [29] L. Nadel, A. Samsonovich, L. Ryan, M. Moscovitch, Multiple trace theory of human memory: computational, neuroimaging, and neuropsychological results, *Hippocampus* 10 (4) (2000) 352–368.

- [30] H. Poon, P. Domingos, Sum-product networks: A new deep architecture, in: Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on, IEEE, 2011, pp. 689–690.

Chapter 3

Holistic Representations for Memorization and Inference

Holistic Representations for Memorization and Inference

Yunpu Ma*
LMU
Siemens AG
Otto-Hahn-Ring 6
81739 Munich

Marcel Hildebrandt
LMU
Siemens AG
Otto-Hahn-Ring 6
81739 Munich

Stephan Baier
LMU
Oettingenstr. 67
80538 Munich

Volker Tresp
LMU
Siemens AG
Otto-Hahn-Ring 6
81739 Munich

Abstract

In this paper we introduce a novel holographic memory model for the distributed storage of complex association patterns and apply it to knowledge graphs. In a knowledge graph, a labelled link connects a subject node with an object node, jointly forming a subject-predicate-objects triple. In the presented work, nodes and links have initial random representations, plus *holistic representations* derived from the initial representations of nodes and links in their local neighbourhoods. A memory trace is represented in the same vector space as the holistic representations themselves. To reduce the interference between stored information, it is required that the initial random vectors should be pairwise quasi-orthogonal. We show that pairwise quasi-orthogonality can be improved by drawing vectors from heavy-tailed distributions, e.g., a Cauchy distribution, and, thus, memory capacity of holistic representations can significantly be improved. Furthermore, we show that, in combination with a simple neural network, the presented holistic representation approach is superior to other methods for link predictions on knowledge graphs.

1 INTRODUCTION

An associative memory is a key concept in artificial intelligence and cognitive neuroscience for learning and memorizing relationships between entities and concepts. Various computational models of associative memory have been proposed, see, e.g., [Hopfield 1982; Gentner 1983]. One important family of associative memory

models is the holographic associative memory (HAM), which was first proposed in [Gabor 1969]. HAMs can store a large number of stimulus-response pairs as additive superpositions of memory traces. It has been suggested that this holographic storage is related to the working principle of the human brain [Westlake 1970].

An important extension to the HAM is based on holographic reduced representations (HRR) [Plate 1995]. In HRR, each entity or symbol is represented as a vector defined in a continuous space. Associations between two entities are compressed in the same vector space via a vector binding operation; the resulting vector is a memory trace. Two associated entities are referred to as a *cue-filler* pair, since a noisy version of the *filler* can be recovered from the memory trace and the *cue* vector via a decoding operation. Multiple *cue-filler* pairs can be compressed in a single memory trace through superposition. Associations can be read out from this single trace, however with large distortions. Thus, a clean-up mechanism was introduced into HRR, such that associations can be *retrieved* with high probability.

The number of associations which can be compressed in a single trace is referred to as *memory capacity*. It has been shown in [Plate 1995] that the memory capacity of the HRR depends on the degree of the pairwise orthogonality of initial random vectors associated with the entities.

Quasi-orthogonality was put forward in [Diaconis et al. 1984; Hall et al. 2005]. They informally stated that “most independent high-dimensional random vectors are nearly orthogonal to each other”. A rigorous mathematical justification to this statement has only recently been given in [Cai et al. 2012; Cai et al. 2013], where the density function of pairwise angles among a large number of Gaussian random vectors was derived. To the best of our knowledge, density functions for other distributions have not been derived, so far. As a first contribution, we will derive a significantly improved quasi-orthogonality, and

*yunpu.ma@siemens.com

we show that memory capacity of holographic representations can significantly be improved. Our result could potentially have numerous applications, e.g., in sparse random projections or random geometric graphs [Penrose 2003].

After the HRR had been proposed, it had mainly been tested on small toy datasets. Quasi-orthogonality becomes exceedingly important when a large amount of entities needs to be initialized with random vectors, as in applications involving large-scale knowledge graphs.

Modern knowledge graphs (KGs), such as FREEBASE [Bollacker et al. 2008], YAGO [Suchanek et al. 2007], and GDELT [Leetaru et al. 2013], are relational knowledge bases, where nodes represent entities and directed labelled links represent predicates. An existing labelled link between a head node (or subject) and a tail node (or object) is a triple and represents a fact, e.g. (*California, locatedIn, USA*).

As a second contribution, we demonstrate how the holographic representations can be applied to KGs. First, one needs to define association pairs (or *cue-filler* pairs). We propose that the representation of a *subject* should encode all *predicate-object* pairs, such that given the *predicate* representation as a *cue*, the *object* should be recovered or at least recognized. Similarly, the representation of an *object* should encode all *predicate-subject* pairs, such that the *subject* can be retrieved after decoding with the *predicate* representation. We call those representations *holistic*, since they are inspired by the semantic holism in the philosophy of language, in the sense that an abstract entity can only be comprehended through its relationships to other abstract entities.

So far we have discussed memory formation and memory retrieval. Another important function is the generalization of stored memory to novel facts. This has technical applications and there are interesting links to human memory. From a cognitive neuroscientist point of view, the brain requires a dual learning system: one is the hippocampus for rapid memorization, and the other is the neocortex for gradual consolidation and comprehension. This hypothesis is the basis for the *Complementary Learning System* (CLS) which was first proposed in [McClelland et al. 1995]. Connections between KGs and long-term declarative memories has recently been stated in [Tresp et al. 2017a; Ma et al. 2018; Tresp et al. 2017b].

As a third contribution of this paper, we propose a model which not only memorizes patterns in the training datasets through holistic representations, but also is able to infer missing links in the KG, by a simple neural network that uses the holistic representations as input representations. Thus, our model realizes a form of

a *complementary learning system*. We compare our results on multiple datasets with other state-of-the-art link prediction models, such as RESCAL [Nickel et al. 2011; Nickel et al. 2012], DISTMULT [Yang et al. 2014], COMPLEX [Trouillon et al. 2016], and R-GCN [Schlichtkrull et al. 2018].

The above mentioned learning-based methods model the KGs by optimizing the latent representations of entities and predicates through minimizing the loss function. It had been observed that latent embeddings are suitable for capturing global connectivity patterns and generalization [Nickel et al. 2016a; Toutanova et al. 2015], but are not as good in memorizing unusual patterns, such as patterns associated with locally and sparsely connected entities. This motivates us to *separate* the memorization and inference tasks. As we will show in our experiments, our approach can, on the one hand, memorize local graph structures, but, on the other hand, also generalizes well to global connectivity patterns, as required by complementary learning systems.

Note, that in our approach holistic representations are derived from random vectors and are **not** learned from data via backpropagation, as in most learning-based approaches to representation learning on knowledge graphs. One might consider representations derived from random vectors to be biologically more plausible, if compared to representations which are learned via complex gradient based update rules [Nickel et al. 2016a]. Thus, in addition to its very competitive technical performance, one of the interesting aspects of our approach is its biological plausibility.

In Section 2 we introduce notations for KGs and embedding learning. In Section 3 we discuss improved quasi-orthogonality by using heavy-tailed distributions. In Section 4 we propose our own algorithm for holistic representations, and test it on various datasets. We also discuss how the memory capacity can be improved. In Section 5 we propose a model which can infer implicit links on KGs through holistic representations. Section 6 contains our conclusions.

2 REPRESENTATION LEARNING

In this section we provide a brief introduction to representation learning in KGs, where we adapt the notation of [Nickel et al. 2016b]. Let \mathcal{E} denotes the set of entities, and \mathcal{P} the set of predicates. Let N_e be the number of entities in \mathcal{E} , and N_p the number of predicates in \mathcal{P} .

Given a predicate $p \in \mathcal{P}$, the characteristic function $\phi_p : \mathcal{E} \times \mathcal{E} \rightarrow \{1, 0\}$ indicates whether a triple (\cdot, p, \cdot) is true or false. Moreover, \mathcal{R}_p denotes the set of all subject-object pairs, such that $\phi_p = 1$. The entire KG can be

written as $\chi = \{(i, j, k)\}$, with $i = 1, \dots, N_e$, $j = 1, \dots, N_p$, and $k = 1, \dots, N_e$.

We assume that each entity and predicate has a unique latent representation. Let \mathbf{a}_{e_i} , $i = 1, \dots, N_e$, be the representations of entities, and \mathbf{a}_{p_i} , $i = 1, \dots, N_p$, be the representations of predicates. Note that \mathbf{a}_{e_i} and \mathbf{a}_{p_i} could be real- or complex-valued vectors/matrices.

A probabilistic model for the KG χ is defined as $\Pr(\phi_p(s, o) = 1 | \mathcal{A}) = \sigma(\eta_{spo})$ for all (s, p, o) -triples in χ , where $\mathcal{A} = \{\mathbf{a}_{e_i}\}_i^{N_e} \cup \{\mathbf{a}_{p_i}\}_i^{N_p}$ denotes the collection of all embeddings; $\sigma(\cdot)$ denotes the sigmoid function; and η_{spo} is the a function of latent representations, \mathbf{a}_s , \mathbf{a}_p and \mathbf{a}_o . Given a labeled dataset containing both true and false triples $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^m$, with $x_i \in \chi$, and $y_i \in \{1, 0\}$, latent representations can be learned. Commonly, one minimizes a binary cross-entropy loss

$$-\frac{1}{m} \sum_{i=1}^m (y_i \log(p_i) + (1 - y_i) \log(1 - p_i)) + \lambda \|\mathcal{A}\|_2^2, \quad (1)$$

where m is the number of training samples, and λ is the regularization parameter; $p_i := \sigma(\eta_{x_i})$ with $\sigma(\cdot)$ being the sigmoid function. η_{spo} is defined differently in various models.

For instance, for RESCAL entities are represented as r -dimensional vectors, $\mathbf{a}_{e_i} \in \mathbb{R}^r$, $i = 1, \dots, N_e$, and predicates are represented as matrices, $\mathbf{a}_{p_i} \in \mathbb{R}^{r \times r}$, $i = 1, \dots, N_p$. Moreover, one uses $\eta_{spo} = \mathbf{a}_s^\top \mathbf{a}_p \mathbf{a}_o$.

For DISTMULT, $\mathbf{a}_{e_i}, \mathbf{a}_{p_j} \in \mathbb{R}^r$, with $i = 1, \dots, N_e$, $j = 1, \dots, N_p$; $\eta_{spo} = \langle \mathbf{a}_s, \mathbf{a}_p, \mathbf{a}_o \rangle$, where $\langle \cdot, \cdot, \cdot \rangle$ denotes the tri-linear dot product.

For COMPLEX, $\mathbf{a}_{e_i}, \mathbf{a}_{p_j} \in \mathbb{C}^r$, with $i = 1, \dots, N_e$, $j = 1, \dots, N_p$; $\eta_{spo} = \Re(\langle \mathbf{a}_s, \mathbf{a}_p, \bar{\mathbf{a}}_o \rangle)$, where the bar denotes complex conjugate, and \Re denotes the real part.

3 DERIVATION OF ϵ -ORTHOGONALITY

As we have discussed in the introduction, quasi-orthogonality of the random vectors representing the entities and the predicates is required for low interference memory retrieval. In this section we investigate the asymptotic distribution of pairwise angles in a set of independently and identically drawn random vectors. In particular, we study random vectors drawn from either a Gaussian or a heavy-tailed Cauchy distribution distribution. A brief summary of notations is referred to the A.7. First we define the term “ ϵ -orthogonality”.

Definition 1. A set of n vectors $\mathbf{x}_1, \dots, \mathbf{x}_n$ is said to be pairwise ϵ -orthogonal, if $|\langle \mathbf{x}_i, \mathbf{x}_j \rangle| < \epsilon$ for $i, j = 1, \dots, n$, $i \neq j$.

Here, $\epsilon > 0$ is a small positive number, and $\langle \cdot, \cdot \rangle$ denotes the inner product in the vector space.

3.1 ϵ -ORTHOGONALITY FOR A GAUSSIAN DISTRIBUTION

In this section we revisit the empirical distribution of pairwise angles among a set of random vectors. More specifically, let $\mathbf{X}_1, \dots, \mathbf{X}_n$ be independent q -dimensional Gaussian variables with distribution $\mathcal{N}(\mathbf{0}, \mathbf{I}_q)$. Denote with Θ_{ij} the angle between \mathbf{X}_i and \mathbf{X}_j , and $\rho_{ij} := \cos \Theta_{ij} \in [-1, 1]$. [Cai et al. 2012; Muirhead 2009] derived the density function of ρ_{ij} in the following Lemma.

Lemma 1. Consider ρ_{ij} as defined above. Then $\{\rho_{ij} | 1 < i < j \leq n\}$ are pairwise i.i.d. random variables with the following asymptotic probability density function

$$g(\rho_G) = \frac{1}{\sqrt{\pi}} \frac{\Gamma(\frac{q}{2})}{\Gamma(\frac{q-1}{2})} (1 - \rho_G^2)^{\frac{q-3}{2}}, \quad |\rho_G| < 1, \quad (2)$$

with fixed dimensionality q .

[Cai et al. 2013] also derived the following Theorem 1.

Theorem 1. Let the empirical distribution μ_n of pairwise angles Θ_{ij} , $1 \leq i < j \leq n$ be defined as $\mu_n := \frac{1}{\binom{n}{2}} \sum_{1 \leq i < j \leq n} \delta_{\Theta_{ij}}$. With fixed dimension q , as $n \rightarrow \infty$, μ_n converges weakly to the distribution with density

$$h(\theta) = \frac{1}{\sqrt{\pi}} \frac{\Gamma(\frac{q}{2})}{\Gamma(\frac{q-1}{2})} (\sin \theta)^{q-2}, \quad \theta \in [0, \pi]. \quad (3)$$

From the above distribution function we can derive the upper bound of quasi-orthogonal random vectors with pairwise ϵ -orthogonality in the Euclidean space \mathbb{R}^q .

Corollary 1. Consider a set of independent q -dimensional Gaussian random vectors which are pairwise ϵ -orthogonal with probability $1 - \nu$, then the number of such Gaussian random vectors is bounded by

$$N \leq \sqrt[4]{\frac{\pi}{2q}} e^{\frac{\epsilon^2 q}{4}} \left[\log \left(\frac{1}{1 - \nu} \right) \right]^{\frac{1}{2}}. \quad (4)$$

The derivation is given in A.1. Due to the symmetry of density function $g(\rho_G)$, we immediately have $\mathbb{E}[\rho_G] = 0$, moreover, $\mathbb{E}[\theta] = \frac{\pi}{2}$. However, for the later use, it is important to consider the expected absolute value of ρ_G :

Corollary 2. Consider a set of n q -dimensional random Gaussian vectors, we have

$$\lambda_G := \mathbb{E}[|\rho_G|] = \sqrt{\frac{2}{\pi q}}. \quad (5)$$

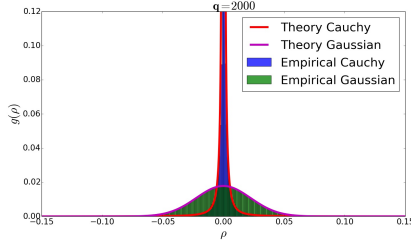


Figure 1: Empirical pairwise angle distribution in a set of Gaussian random vectors (green) is compared with theoretical prediction Eq. 2 (magenta); Empirical pairwise angle distribution in a set of Cauchy random vectors (blue) is compared with prediction Eq. 6 (red)

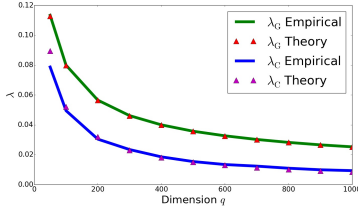


Figure 2: Compare λ_G and λ_C from simulation and theory, see Eq. 5 and Eq. 9.

Note, that the quantity $\frac{\pi}{2} - \arccos \mathbb{E}[|\rho_G|]$ has a clear geometrical meaning: It indicates the expected deviation from $\frac{\pi}{2}$ of pairwise angles. In fact, in the extreme case when $q \rightarrow \infty$, the deviation converges to 0 with the rate \sqrt{q} .

3.2 ϵ -ORTHOGONALITY FOR A CAUCHY DISTRIBUTION

In this subsection, we show that the set of random vectors whose elements are initialized with a heavy-tailed distribution, e.g., a Cauchy distribution $\mathcal{C}(0, 1)$, has improved ϵ -orthogonality. The intuition is as follows: Consider a set of q -dimensional random vectors initialized with a heavy-tailed distribution. After normalization, each random vector can be approximated by only the elements which significantly deviate from zero and were drawn from the heavy tails. If the number of those elements is k with $k \ll q$, then there are at most $\binom{q}{k}$ orthogonal random vectors.

Moreover, $\binom{q}{k} \approx \frac{q^k}{k! \Gamma(k)}$ could be much larger than $\sqrt[4]{\frac{\pi}{2q}} e^{\frac{c^2 q}{4}}$ from Eq. 4, when q is sufficiently large, $k \ll q$, and $\epsilon \rightarrow 0$. In other words, under stricter quasi-orthogonality condition with smaller ϵ , random vectors drawn from a heavy-tailed distribution could have more pairs satisfying the quasi-orthogonality condition.

Consider a set of q -dimensional Cauchy random vectors. As $q \rightarrow \infty$ the approximate density function of ρ_{ij} , with $1 \leq i < j \leq n$ is described in the following conjecture.

Conjecture 1. Let $\mathbf{X}_1, \dots, \mathbf{X}_n$ be independent q -dimensional random vectors whose elements are independently and identically drawn from Cauchy a distribution $\mathcal{C}(0, 1)$. Moreover, consider the angle Θ_{ij} between \mathbf{X}_i , and \mathbf{X}_j . Then, as $q \rightarrow \infty$, $\rho_{ij} := \cos \Theta_{ij} \in [-1, 1]$, $1 \leq i < j \leq n$ are pairwise i.i.d. with a density function approximated by

$$g(\rho_C) = -\frac{2}{\pi^2 q^2 \rho_C^3} \cdot \frac{1}{z^{\frac{3}{2}}} \left[e^{\frac{1}{\pi z}} \text{Ei} \left(-\frac{1}{\pi z} \right) \right], \quad (6)$$

where $z := \frac{1}{q^2} \left(\frac{1}{\rho_C^2} - 1 \right)$, and the exponential integral $\text{Ei}(x)$ is defined as $\text{Ei}(x) = -\int_{-x}^{\infty} \frac{e^{-t}}{t} dt$.

The intuition behind the conjecture is as follows. Suppose $\mathbf{X} = (X_1, \dots, X_q)$ and $\mathbf{Y} = (Y_1, \dots, Y_q)$ are random vector variables, and assume that elements of \mathbf{X} and \mathbf{Y} are independently Gaussian distributed. In order to derive $g(\rho_{\mathbf{X}, \mathbf{Y}})$ in Lemma 1, [Cai et al. 2012; Muirhead 2009] compute the distribution function for $\frac{\alpha^T \mathbf{X}}{\|\mathbf{X}\|}$ instead, where $\alpha^T \alpha = 1$. In particular, they assume that $\alpha = (1, 0, \dots, 0)$. The underlying reason for this assumption is that the random vector $\frac{\mathbf{X}}{\|\mathbf{X}\|}$ is uniformly distributed on the $(q-1)$ -dimensional sphere.

Here, elements of \mathbf{X} and \mathbf{Y} are independently Cauchy distributed. We derive the approximation in Eq. 6 under the same assumption by taking $g(\rho_{\mathbf{X}, \mathbf{Y}}) \approx \frac{X_1}{\sqrt{X_1^2 + \dots + X_q^2}}$. Furthermore, we introduce a new variable $z_{\mathbf{X}, \mathbf{Y}} := \frac{1}{q^2} \left(\frac{1}{\rho_{\mathbf{X}, \mathbf{Y}}^2} - 1 \right) = \frac{1}{q^2} \frac{X_2^2 + \dots + X_q^2}{X_1^2}$, and derive the density function $\hat{g}(z_{\mathbf{X}, \mathbf{Y}})$ by using the generalized central limit theorem [Gnedenko et al. 1954] and properties of quotient distributions of two independent random variables. $g(\rho_{\mathbf{X}, \mathbf{Y}})$ can be directly obtained from $\hat{g}(z_{\mathbf{X}, \mathbf{Y}})$ by a variable transform. More details and derivation are referred to the A.2.

We turn to study the limiting behaviour of the density function when ρ approaches zero. In this case, the variable z defined in in Conjecture 1 can be approximated by $z \approx \frac{1}{q^2 \rho_C^2}$. Using properties of the exponential integral, as $q \rightarrow \infty$, the density function in Eq. 6 can be approximated by its Laurent series,

$$g(\rho_C) \approx \frac{2}{\pi q \rho_C^2} - \frac{2}{q^3 \rho_C^4} + \frac{4\pi}{q^5 \rho_C^6} + \mathcal{O} \left(\frac{1}{q^7 \rho_C^8} \right) \quad (7)$$

In the following corollary we give the upper bound of the number of pairwise ϵ -orthogonal Cauchy random vectors using Eq. 6.

Corollary 3. Consider a set of independent q -dimensional Cauchy random vectors which are pairwise ϵ -orthogonal with probability $1 - \nu$, then the number of such Cauchy random vectors is bounded by

$$N \leq \sqrt{\frac{\pi \epsilon q}{4}} \left[\log \left(\frac{1}{1 - \nu} \right) \right]^{\frac{1}{2}}. \quad (8)$$

Let us compare the prefactor of this upper bound for two distributions: That is $\sqrt{\frac{\pi \epsilon q}{4}} e^{\frac{\epsilon^2 q}{4}}$ for the Gaussian distribution, and $\sqrt{\frac{\pi \epsilon q}{4}}$ for the Cauchy distribution. Under strict quasi-orthogonal conditions with arbitrarily small but fixed $\epsilon > 0$, for the dimension $q \gg 2\sqrt{\frac{1}{\pi \epsilon^2}}$ we have that $\sqrt{\frac{\pi \epsilon q}{4}} \gg \sqrt[4]{\frac{\pi}{2q}} e^{\frac{\epsilon^2 q}{4}} \approx \sqrt[4]{\frac{\pi}{2q}}$. It implies that in sufficiently high-dimensional spaces, random vectors which are independently drawn from a Cauchy distribution are more likely to satisfy the pairwise ϵ -orthogonality condition - particularly when $\epsilon \ll 1$.

Remark 1. For the later use, we define λ_C as $\lambda_C := \mathbb{E}[|\rho_C|]$ for the case of Cauchy distribution. However, no simple analytic form is known for this integral. Thus we use the following numerically stable and non-divergent equation to approximate λ_C ,

$$\lambda_C \approx -\frac{4q}{\pi^2} \int_0^1 \rho \left[e^{\frac{q^2 \rho^2}{\pi}} \text{Ei} \left(-\frac{q^2 \rho^2}{\pi} \right) \right] d\rho. \quad (9)$$

This simpler form is derived from Eq. 6 using the approximation $z \approx \frac{1}{q^2 \rho^2}$.

Fig. 1 shows the empirical distribution of ρ_G in a set of Gaussian random vectors (green) compared with theoretical prediction in Eq.2 (magenta); and the empirical distribution of ρ_C in a set of Cauchy random vectors (blue) compared with theoretical prediction (red). In the case of Cauchy random vectors, the leading orders of the Laurent expansion of Eq. 6 are used, see Eq. 7. For the empirical simulation, 10000 random vectors with dimensionality $q = 2000$ were drawn independently from either a Gaussian or a Cauchy distribution.

In addition, in Fig. 2 we plot λ_G and λ_C as a function of q in comparison with the theoretical predictions from Eq. 5 and Eq. 9, respectively, under the same simulation condition. It is necessary to emphasize that $\lambda_C(q) < \lambda_G(q)$ for all the dimensions q ; this fact will be used to explain the relatively high memory capacity encoded from Cauchy random vectors.

In the Appendix, see Remark A 2, the distribution of elements from the normalized random variable $\frac{\mathbf{x}}{\|\mathbf{x}\|}$ is also considered. In particular, for normalized Cauchy random vector most of its elements are nearly zero, and it realizes a **sparse** representation.

4 HOLISTIC REPRESENTATIONS FOR KGS

4.1 HRR MODEL

First, we briefly review HRR. Three operations are defined in HRR to model associative memories: *encoding*, *decoding*, and *composition*.

Let \mathbf{a} , \mathbf{b} , \mathbf{c} , and \mathbf{d} be random vectors representing different entities. The encoding phase stores the association between \mathbf{a} and \mathbf{b} in a memory trace $\mathbf{a} * \mathbf{b}$, where $*$: $\mathbb{R}^q \times \mathbb{R}^q \rightarrow \mathbb{R}^q$ denotes circular convolution, which is defined as $[\mathbf{a} * \mathbf{b}]_k = \sum_{i=0}^{q-1} a_i b_{(k-i) \bmod q}$.

A noisy version of \mathbf{b} can be retrieved from the memory trace, using the item \mathbf{a} as a cue, with: $\mathbf{b} \approx \mathbf{a} * (\mathbf{a} * \mathbf{b})$, where $*$: $\mathbb{R}^q \times \mathbb{R}^q \rightarrow \mathbb{R}^q$ denotes the circular correlation¹. It is defined as $[\mathbf{a} * \mathbf{b}]_k = \sum_{i=0}^{q-1} a_i b_{(k+i) \bmod q}$. In addition, several associations can be superimposed in a single trace via the addition operation: $(\mathbf{a} * \mathbf{b}) + (\mathbf{c} * \mathbf{d}) + \dots$.

4.2 HOLISTIC MODEL

Initially, each entity and predicate in a KG is associated with a q -dimensional normalized random vector, which is then normalized. We denote them as $\mathbf{r}_{e_i}^{G/C}$, $i = 1, \dots, N_e$, and $\mathbf{r}_{p_i}^{G/C}$, $i = 1, \dots, N_p$, respectively. The superscript indicates from which distribution vector elements are independently drawn, either the Gaussian or Cauchy distribution. If there is no confusion, we may omit the superscript.

Consider an entity e_i . Let $\mathcal{S}^s(e_i) = \{(p, o) | \phi_p(e_i, o) = 1\}$ be the set of all predicate-object pairs for which triples (e_i, p, o) is true and where e_i is the subject. We store these multiple associations in a single memory trace via circular correlation and superposition:

$$\mathbf{h}_{e_i}^s = \sum_{(p, o) \in \mathcal{S}^s(e_i)} [\text{Norm}(\mathbf{r}_p * \mathbf{r}_o) + \xi \mathbf{r}_{e_i}], \quad (10)$$

where $\text{Norm} : \mathbb{R}^q \rightarrow \mathbb{R}^q$ represents the normalization operation², which is defined as $\text{Norm}(\mathbf{r}) := \frac{\mathbf{r}}{\|\mathbf{r}\|}$. Moreover, the hyper-parameter $\xi > 0$ determines the contribution of the individual initial representation \mathbf{r} .

¹It uses the fact that $\mathbf{a} * \mathbf{a} \approx \delta$, where δ is the identity operation of convolution.

²In other sections, we may obviate Norm operator in the equation for the sake of simplicity, since it can be shown that the circular correlation of two normalized high-dimensional random vectors are almost normalized.

Note, that the same entity e_i could also play the role of an object. For instance, the entity *California* could be the subject in the triple (*California, locatedIn, USA*), or the object in another triple (*Paul, livesIn, California*). Thus, it is necessary to have another representation to specify its role in the triples. Consider the set of subject-predicate pairs $\mathcal{S}^o(e_i) = \{(s, p) | \phi_p(s, e_i) = 1\}$ for which triples (s, p, e_i) are true. These pairs are stored in a single trace via $\mathbf{h}_{e_i}^o = \sum_{(s, p) \in \mathcal{S}^o(e_i)} [\text{Norm}(\mathbf{r}_p \star \mathbf{r}_s) + \xi \mathbf{r}_{e_i}]$, where $\mathbf{h}_{e_i}^o$ is the representation of the entity e_i when it acts as an object.

For the later generalization task, the overall holistic representation for the entity e_i is defined as the summation of both representations, namely

$$\mathbf{h}_{e_i} = \mathbf{h}_{e_i}^s + \mathbf{h}_{e_i}^o. \quad (11)$$

In this way, the complete neighbourhood information of an entity can be used for generalization.

Furthermore, given a predicate p_i , the holistic representation \mathbf{h}_{p_i} encodes all the subject-object pairs in the set $\mathcal{S}(p_i) = \{(s, o) | \phi_{p_i}(s, o) = 1\}$ via

$$\mathbf{h}_{p_i} = \sum_{(s, o) \in \mathcal{S}(p_i)} [\text{Norm}(\mathbf{r}_s \star \mathbf{r}_o) + \xi \mathbf{r}_{p_i}]. \quad (12)$$

After storing all the association pairs into holistic features of entities and predicates, the initial randomly assigned representations are not required anymore and can be deleted. These representations are then fixed and not trainable unlike other embedding models.

After encoding, entity retrieval is performed via a circular convolution. Consider a concrete triple (e_1, p_1, e_2) with unknown e_2 . The identity of e_2 could be revealed with the holistic representation of p_1 and the holistic representation of e_1 as a subject, namely \mathbf{h}_{p_1} and $\mathbf{h}_{e_1}^s$. Then retrieval is performed as $\mathbf{h}_{p_1} \star \mathbf{h}_{e_1}^s$. The associations can be retrieved from the holography memory with low fidelity due to interference. Therefore, after decoding, a clean-up operation is employed, as in the HRR model. Specifically, a nearest neighbour is determined using cosine similarity. The pseudo-code for encoding holistic representations is provided in A.6.

4.3 EXPERIMENTS ON MEMORIZATION

We test the memorization of complex structure on different datasets and compare the performance of different models. Recall that \mathcal{R}_p is the set of all true triples with respect to a given predicate p . Consider a possible triple $(s, p, o) \in \mathcal{R}_p$. The task is now to retrieve the object entity from holistic vectors \mathbf{h}_s and \mathbf{h}_p , and to retrieve the subject entity from holistic vectors \mathbf{h}_p and \mathbf{h}_o .

As discussed, in retrieval, the noisy vector $\mathbf{r}'_o = \mathbf{h}_p \star \mathbf{h}_s$ is compared to the holistic representations of all entities using cosine similarity, according to which the entities are then ranked. In general, multiple objects could be connected to a single subject-predicate pair. Thus, we employ the *filtered mean rank* introduced in [Bordes et al. 2013] to evaluate the memorization task.

We have discussed that the number of pairwise quasi-orthogonal vectors crucially depends on the random initialization. Now we analyse, if the memory capacity depends on the quasi-orthogonality of the initial representation vectors, as well. We perform memorization task on three different KGs, which are FB15k-237 [Toutanova et al. 2015], YAGO3 [Mahdisoltani et al. 2013], and a subset of GDEL T [Leetaru et al. 2013]. The exact statistics of the datasets are given in Table. 1.

Table 1: Statistics of KGs

	$\#\mathcal{D}$	N_a	N_e	N_p
GDEL T	497,605	≈ 73	6786	231
FB15k-237	301,080	≈ 20	14505	237
YAGO3	1,089,000	≈ 9	123143	37

Recall that N_e and N_p denote the number of entities and predicates, respectively. Moreover, $\#\mathcal{D}$ denotes the total number of triples in a KG, and N_a is the average number of association pairs compressed into holistic feature vectors of entities, which can be estimated as $\frac{\#\mathcal{D}}{N_e}$. After encoding triples in a dataset into holistic features, filtered mean rank is evaluated by ranking retrieved subjects and objects of all triples. Filtered mean ranks on three datasets with holistic representations encoded from Gaussian and Cauchy distributions are displayed in Fig. 3 (a)-(c).

Cauchy holistic representations outperform Gaussian holistic representations significantly when the total number of entities is large (see, Fig. 3(c) for YAGO3), or the average number of encoded associations is large (see, Fig. 3(a) for GDEL T). This implies that quasi-orthogonality plays an important role in holographic memory. Improved quasi-orthogonality allows for more entities to be initialized with quasi-orthogonal representations, which is very important for memorizing huge KGs. In addition, it reduces the interference between associations. Moreover, Cauchy holistic features are intrinsically very sparse, making them an attractive candidate for modeling biologically plausible memory systems.

4.4 CORRELATION VERSUS CONVOLUTION

One of the main differences between *holistic representation* and the *holographic reduced representation* is the binding operation. In HRR, two vectors are composed

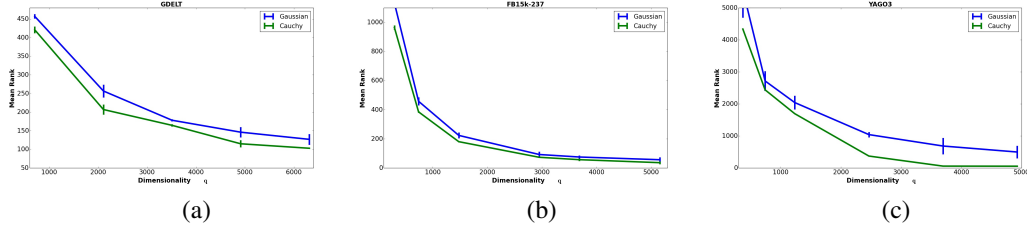


Figure 3: Filtered MR vs. the dimensionality of holistic representations evaluated on dataset: (a) GDELT, (b) FB15k-237, and (c) YAGO3. Blues lines denote holistic representations encoded from Gaussian random vectors, and green lines denote holistic representations encoded from Cauchy random vectors. Lower values are preferred.

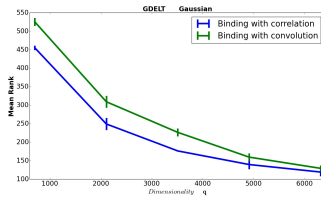


Figure 4: Filtered MR vs. the dimensionality of holistic representations evaluated on the GDELT dataset with Gaussian initialization.

via circular convolution, while in holistic representation, they are composed via circular correlation.

Binding with convolution and correlation is compared in Fig. 4. We report the filtered MR scores on the GDELT dataset versus the dimensionality of holistic representations. It can be seen that binding with circular correlation is significantly superior to convolution. Therefore, a non-commutative compositional operator is essential for storing the directed structures of KG into holographic memory. A theoretical explanation is given in the A.4, along with experimental results on other datasets.

4.5 HYPER-PARAMETER ξ

In the experiments so far, the optimal hyper-parameter ξ is found via grid search. However, it is possible to roughly estimate the range of the optimal hyper-parameter ξ . Indeed, ξ strongly depends on λ_G or λ_C and the average number of encoded association pairs N_a .

So far, the deep relation between holographic memory capacity and quasi-orthogonality has not been discussed in the literature. In the original work on HRR, memory capacity and information retrieval quality are estimated from the distribution of elements in random vectors. In this section we give a plausible explanation from the point of view of the pairwise angle distribution.

Consider a subject s . The predicate-object pair (p, o)

is stored in the holistic representation \mathbf{h}_s along with the other $N_a - 1$ pairs, such that

$$\mathbf{h}_s = \xi N_a \mathbf{r}_s + \mathbf{r}_p \star \mathbf{r}_o + \sum_{i=2}^{N_a} \mathbf{r}_{p_i} \star \mathbf{r}_{o_i}.$$

Suppose we try to identify the object in the triple (s, p, \cdot) via \mathbf{h}_s and \mathbf{h}_p . After decoding, the noisy vector $\mathbf{r}'_o = \mathbf{h}_p \star \mathbf{h}_s$ should be recalled with \mathbf{h}_o , which is the holistic representation of o . Let $\theta_{\mathbf{r}'_o, \mathbf{h}_o}$ denote the angle between \mathbf{r}'_o and \mathbf{h}_o . The cosine function of this angle is again defined as $\rho_{\mathbf{r}'_o, \mathbf{h}_o} := \cos \theta_{\mathbf{r}'_o, \mathbf{h}_o}$.

In order to recall the object successfully, the angle between \mathbf{r}'_o and \mathbf{h}_o should be smaller than the expected absolute angle between two arbitrary vectors, namely

$$\theta_{\mathbf{r}'_o, \mathbf{h}_o} < \mathbb{E}[\|\theta_{G/C}\|], \quad (13)$$

This inequality first implies that the optimal ξ should be a positive number. Given the definition of $\lambda_{G/C}$ in Eq. 5 and 9, equivalently, Eq. 13 requires

$$\rho_{\mathbf{r}'_o, \mathbf{h}_o} > \lambda_{G/C}. \quad (14)$$

After some manipulations, a sufficient condition to recognize the object correctly is given by (see A.5)

$$\begin{aligned} \rho_{\mathbf{r}'_o, \mathbf{h}_o} &> \frac{\xi^2 N_a^2 - (\xi^3 N_a^3 + 2\xi^2 N_a^3 - \xi^2 N_a^2 + \xi N_a^2 + \xi N_a^3) \lambda_{G/C}}{\xi^2 N_a^2 + N_a + 2\xi N_a^2 \lambda_{G/C} + N_a(N_a - 1) \lambda_{G/C}} \\ &> \lambda_{G/C}. \end{aligned} \quad (15)$$

In the following, we verify this condition on the FB15k-237 dataset. We consider one of the experimental settings employed in the memorization task. The dimension of holistic features is $q = 5200$, with $\lambda_G = 0.0111$ computed from Eq. 5, and $\lambda_C = 0.00204$ from Eq. 9. For Gaussian initialization, the optimum is found at $\xi = 0.14$ via grid search, while for Cauchy initialization, the optimum is found at $\xi = 0.05$.

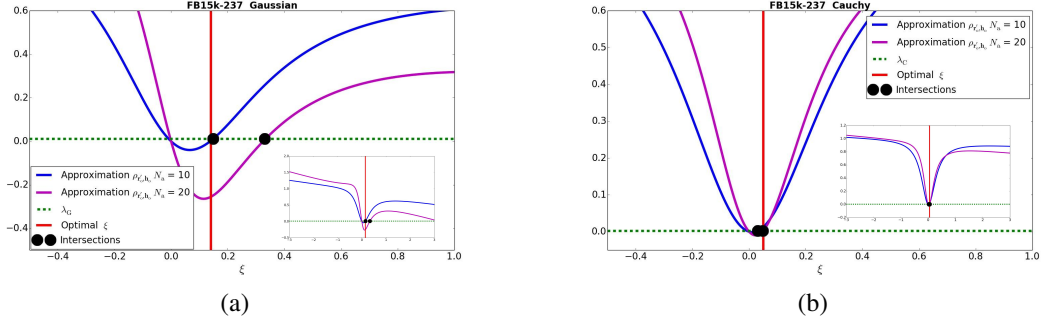


Figure 5: Analysis of the hyper-parameter ξ on the FB15k-237 dataset. (a): Approximation of $\rho_{r'_o, h_o}$ for Gaussian initialization. Curves with $N_a = 10$ (blue), $N_a = 20$ (magenta) and their intersections with the retrieval threshold λ_G are displayed. The red vertical line denotes the experimentally determined optimal ξ . Insert shows the curves with $\xi \in [-3, 3]$. (b): Approximation of $\rho_{r'_o, h_o}$ for Cauchy initialization with $N_a = 10$ (blue), and $N_a = 20$ (magenta). Rest remains the same.

To verify these optima, Fig. 5 (a) and (b) display the approximation of $\rho_{r'_o, h_o}(\xi, N_a)$ as a function of ξ .³ Its intersection with $\lambda_{G/C}$ is marked with a black dot. In FB15k-237, N_a is estimated to be 20, while, in general, a KG could be quite imbalanced. Thus, $\rho_{r'_o, h_o}(\xi, N_a)$ with $N_a = 10$, and 20 are shown together for comparison.

In Fig. 5 (a) for Gaussian initialization, experimentally determined optimal ξ (red vertical line) is found close to the intersection of $\rho_{r'_o, h_o}(\xi, N_a = 10)$ and threshold λ_G , meaning that Gaussian holistic features tend to memorize fewer association pairs. They can only map sparsely connected graph structures into meaningful representations.

In Fig. 5 (b) for Cauchy initialization, however, the optimal ξ is close to the intersection of $\rho_{r'_o, h_o}(\xi, N_a = 20)$ and λ_C . Thus, Cauchy holistic features are more suitable to memorize a larger chunk of associations, meaning that they are capable of mapping densely connected graph structures into meaningful representations. All optima are found near the intersection points instead of the local maximum with $\xi > 0$. It indicates that, to maximize the memory capacity, the holistic features can only store information with very low fidelity.

Table 2: Filtered recall scores on FB15k-237

Methods	MR	MRR	Hits		
			@10	@3	@1
RESCAL	996	0.221	0.363	0.237	0.156
DISTMULT	254	0.241	0.419	0.263	0.155
COMPLEX	339	0.247	0.428	0.275	0.158
R-GCN ⁴	-	0.248	0.414	0.258	0.153
HOLNN _G ⁵	235	0.285	0.455	0.315	0.207
HOLNN _C	228	0.295	0.465	0.320	0.212

³The approximation of $\rho_{r'_o, h_o}$ is the second term of Eq. 15

5 INFERENCE ON KG

5.1 INFERENCE VIA HOLISTIC REPRESENTATION

In this section, we describe the model for inferring the missing links in the KG. Recall the scoring function η_{spo} defined in Sec. 2. Our model uses holistic representations as input and generalizes them to implicit facts, by a two-layer neural network⁶. Formally, the scoring function is given as follow:

$$\eta_{spo} = \langle \text{ReLU}(\mathbf{h}_s \mathbf{W}_1^e) \mathbf{W}_2^e, \text{ReLU}(\mathbf{h}_p \mathbf{W}_1^p) \mathbf{W}_2^p, \text{ReLU}(\mathbf{h}_o \mathbf{W}_1^e) \mathbf{W}_2^e \rangle, \quad (16)$$

where $\langle \cdot, \cdot, \cdot \rangle$ denotes tri-linear dot product; $\mathbf{h}_s, \mathbf{h}_o$ are the holistic representations for entities defined in Eq. 11, \mathbf{h}_p is defined in Eq. 12.

Suppose that the holistic representations are defined in \mathbb{R}^q . Then $\mathbf{W}_1^e \in \mathbb{R}^{q \times h_1}$ and $\mathbf{W}_2^e \in \mathbb{R}^{h_1 \times h_2}$ are shared weights for entities; $\mathbf{W}_1^p \in \mathbb{R}^{q \times h_1}$ and $\mathbf{W}_2^p \in \mathbb{R}^{h_1 \times h_2}$ are shared weights for predicates. We refer Eq. 16 as HOLNN, a combination of holistic representations and a simple neural network.

As an example, for training on FB15k-237, we take $q = 3600$, $h_1 = 64$, and $h_2 = 256$. Note that only weight matrices in the neural network are trainable parameters, holistic representations are fixed after encoding. Thus, the total number of trainable parameters in HOLNN is $0.48M$, which is much smaller than COM-

⁴see [Schlichtkrull et al. 2018]

⁵G stands for Gaussian holistic features, and C for Cauchy holistic features.

⁶Further experimental details are referred to A.8

PLEX with 5.9M parameters, by assuming that the dimension of embeddings in the COMPLEX is 200.

To evaluate the performance of HOLNN for missing links prediction, we compare it to the state-of-the-art models on two datasets: FB15k-237, and GDELT. They were split randomly in training, validation, and test sets. We implement all models with the identical loss function Eq. 1, and minimize the loss on the training set using Adam as the optimization method. Hyper-parameters, e.g., the learning rate, and l_2 regularization, are optimized based on the validation set.

We use filtered MR, filtered mean reciprocal rank (MRR), and filtered Hits at n (Hits@ n) as evaluation metrics [Bordes et al. 2013]. Table 2 and Table 3 report different metrics on the FB15k-237, and GDELT dataset, respectively. It can be seen that HOLNN is superior to all the baseline methods on both datasets with considerably less trainable parameters. Moreover, HOLNN_C consistently outperforms HOLNN_G, indicating that the memory capacity of holistic representations is important for generalization.

Table 3: Filtered recall scores on GDELT

Methods	MR	MRR	Hits		
			@10	@3	@1
RESCAL	212	0.202	0.396	0.225	0.107
DISTMULT	181	0.232	0.451	0.268	0.124
COMPLEX	158	0.256	0.460	0.295	0.146
HOLNN _G	105	0.284	0.457	0.301	0.198
HOLNN _C	102	0.296	0.471	0.315	0.210

5.2 INFERENCE ON NEW ENTITIES

In additional experiments, we show that HOLNN is capable of inferring implicit facts on new entities without re-training the neural network. Experiments are performed on FB15k-237 as follows. We split the entire FB15k-237 dataset \mathcal{D} into \mathcal{D}_{old} and \mathcal{D}_{new} . In \mathcal{D}_{new} , the subjects of triples are new entities which do not show up in \mathcal{D}_{old} , while objects and predicates are already seen in the \mathcal{D}_{old} . Suppose our task is to predict implicit links between new entities (subjects in \mathcal{D}_{new}) and old entities (entities in \mathcal{D}_{old}). Thus, we further split \mathcal{D}_{new} into $\mathcal{D}_{new}^{train}$, $\mathcal{D}_{new}^{valid}$, and \mathcal{D}_{new}^{test} sets.

For embedding models, e.g., COMPLEX, after training on \mathcal{D}_{old} , the most efficient way to solve this task is to adapt the embeddings of new entities on $\mathcal{D}_{new}^{train}$, with fixed embeddings of old entities. On the other hand, for the HOLNN model, new entities obtain their holistic representations via triples in the $\mathcal{D}_{new}^{train}$ set. These holistic features are then fed into the trained two-layer neural network. Table 4 shows filtered recall scores for predict-

ing links between new entities and old entities on \mathcal{D}_{new}^{test} , with the number of new entities in \mathcal{D}_{new} being 300, 600, or 900. COMPLEX and HOLNN with Cauchy holistic features are compared.

There are two settings for the HOLNN_C model. New entities could be encoded either from holistic features of old entities, or from random initializations of old entities⁷. We denote these two cases as HOLNN_C(**h**) and HOLNN_C(**r**), respectively. It can be seen that HOLNN_C(**r**) outperforms HOLNN_C(**h**) only to some degree. It indicates that HOLNN_C is robust to the noise, making it generalizes well.

Table 4: Inference of new entities on FB15k-237

Methods	Number of New Entities					
	300		600		900	
	MR	MRR	MR	MRR	MR	MRR
COMPLEX	262	0.291	265	0.266	286	0.243
HOLNN _C (h)	345	0.274	415	0.242	510	0.222
HOLNN _C (r)	252	0.315	302	0.281	395	0.265

6 CONCLUSION

We have introduces the holistic representation for the distributed storage of complex association patterns and have applied it to knowledge graphs. We have shown that interference between stored information is reduced with initial random vectors which are pairwise quasi-orthogonal and that pairwise quasi-orthogonality can be improved by drawing vectors from heavy-tailed distributions, e.g., a Cauchy distribution. The experiments demonstrated excellent performance on memory retrieval and competitive results on link prediction.

In our approach, latent representations are derived from random vectors and are not learned from data, as in most modern approaches to representation learning on knowledge graphs. One might consider representations derived from random vectors to be biologically more plausible, if compared to representations which are learned via complex gradient based update rules. Thus in addition to its very competitive technical performance, one of the interesting aspects of our approach is its biological plausibility.

Outlook: Potential applications could be applying the holistic encoding algorithm to Lexical Functional for modeling distributional semantics [Coecke et al. 2010], or graph convolutional network [Kipf et al. 2017] for semi-supervised learning using holistic representations as feature vectors of nodes on a graph.

⁷Recall that random initializations are actually deleted after encoding. Here we use them just for comparison.

References

- Bollacker, Kurt, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor (2008). “Freebase: a collaboratively created graph database for structuring human knowledge”. *Proceedings of the 208 ACM SIGMOD*. AcM, pp. 1247–1250.
- Bordes, Antoine, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko (2013). “Translating embeddings for modeling multi-relational data”. *NIPS*, pp. 2787–2795.
- Cai, Tony and Tiefeng Jiang (2012). “Phase transition in limiting distributions of coherence of high-dimensional random matrices”. *Journal of Multivariate Analysis* 107, pp. 24–39.
- Cai, Tony, Jianqing Fan, and Tiefeng Jiang (2013). “Distributions of angles in random packing on spheres”. *The Journal of Machine Learning Research* 14.1, pp. 1837–1864.
- Coecke, Bob, Mehrnoosh Sadrzadeh, and Stephen Clark (2010). “Mathematical foundations for a compositional distributional model of meaning”. *Linguistic Analysis* 36.
- Diaconis, Persi and David Freedman (1984). “Asymptotics of graphical projection pursuit”. *The annals of statistics*, pp. 793–815.
- Gabor, D. (1969). “Associative holographic memories”. *IBM Journal of Research and Development* 13.2, pp. 156–159.
- Gentner, Dedre (1983). “Structure-mapping: A theoretical framework for analogy”. *Cognitive science* 7.2, pp. 155–170.
- Gnedenko, B.V. and A.N. Kolmogorov (1954). *Limit distributions for sums of independent random variables*. Addison-Wesley.
- Hall, Peter, James Stephen Marron, and Amnon Neeman (2005). “Geometric representation of high dimension, low sample size data”. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 67.3, pp. 427–444.
- Hopfield, John J. (1982). “Neural networks and physical systems with emergent collective computational abilities”. *Proceedings of the national academy of sciences* 79.8, pp. 2554–2558.
- Kipf, Thomas N and Max Welling (2017). “Semi-supervised classification with graph convolutional networks”. *ICLR*.
- Leetaru, Kalev and Philip A. Schrodt (2013). “GDELT: Global data on events, location, and tone”. *ISA Annual Convention*.
- Ma, Yunpu, Volker Tresp, and Erik Daxberger (2018). “Embedding models for episodic memory”. *arXiv preprint arXiv:1807.00228*.
- Mahdisoltani, Farzaneh, Joanna Biega, and Fabian M. Suchanek (2013). “Yago3: A knowledge base from multilingual wikipedias”. *CIDR*.
- McClelland, James L., Bruce L. McNaughton, and Randall C. O’reilly (1995). “Why there are complementary learning systems in the hippocampus and neocortex: insights from the successes and failures of connectionist models of learning and memory.” *Psychological review* 102.3, p. 419.
- Muirhead, Robb J. (2009). *Aspects of multivariate statistical theory*. Vol. 197. John Wiley & Sons.
- Nickel, Maximilian, Volker Tresp, and Hans-Peter Kriegel (2011). “A Three-Way Model for Collective Learning on Multi-Relational Data”. *ICML*. Vol. 11, pp. 809–816.
- Nickel, Maximilian, Volker Tresp, and Hans-Peter Kriegel (2012). “Factorizing yago: scalable machine learning for linked data”. *Proceedings of the 21st international conference on World Wide Web*. ACM, pp. 271–280.
- Nickel, Maximilian, Kevin Murphy, Volker Tresp, and Evgeniy Gabrilovich (2016a). “A review of relational machine learning for knowledge graphs”. *Proceedings of the IEEE* 104.1, pp. 11–33.
- Nickel, Maximilian, Lorenzo Rosasco, and Tomaso Poggio (2016b). “Holographic Embeddings of Knowledge Graphs”. *AAAI*, pp. 1955–1961.
- Penrose, Mathew (2003). *Random geometric graphs*. 5. Oxford university press.
- Plate, Tony A. (1995). “Holographic reduced representations”. *IEEE Transactions on Neural networks* 6.3, pp. 623–641.
- Schlichtkrull, Michael, Thomas N Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling (2018). “Modeling relational data with graph convolutional networks”. *European Semantic Web Conference*. Springer, pp. 593–607.
- Suchanek, Fabian M., Gjergji Kasneci, and Gerhard Weikum (2007). “Yago: a core of semantic knowledge”. *Proceedings of the 16th international conference on World Wide Web*. ACM, pp. 697–706.
- Toutanova, Kristina and Danqi Chen (2015). “Observed versus latent features for knowledge base and text inference”. *Proceedings of the 3rd Workshop on Continuous Vector Space Models and their Compositionality*, pp. 57–66.
- Tresp, Volker, Yunpu Ma, Stephan Baier, and Yinchong Yang (2017a). “Embedding learning for declarative memories”. *ESWC*. Springer, pp. 202–216.
- Tresp, Volker and Yunpu Ma (2017b). “The Tensor Memory Hypothesis”. *arXiv preprint arXiv:1708.02918*.
- Trouillon, Théo, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard (2016). “Com-

plex embeddings for simple link prediction”. *ICML*, pp. 2071–2080.

Westlake, Philip R. (1970). “The possibilities of neural holographic processes within the brain”. *Kybernetik* 7.4, pp. 129–153.

Yang, Bishan, Wentau Yih, Xiaodong He, Jianfeng Gao, and Li Deng (2014). “Embedding entities and relations for learning and inference in knowledge bases”. *ICLR 2015*.

A APPENDIX

A.1 DERIVATION OF COROLLARY 1 & 2

Corollary 1. Consider a set of independent q -dimensional Gaussian random vectors which are pairwise ϵ -orthogonal with probability $1-\nu$, then the number of such Gaussian random vectors is bounded by

$$N \leq \sqrt[4]{\frac{\pi}{2q}} e^{\frac{\epsilon^2 q}{4}} \left[\log \left(\frac{1}{1-\nu} \right) \right]^{\frac{1}{2}}. \quad (\text{A.1})$$

Proof. Recall that, in the case of Gaussian distributed random vectors, the pdf of ρ is

$$g(\rho) = \frac{1}{\sqrt{\pi}} \frac{\Gamma(\frac{q}{2})}{\Gamma(\frac{q-1}{2})} (1-\rho^2)^{\frac{q-3}{2}}.$$

This directly yields that $\omega := \sqrt{q}\rho$ has the density function

$$f(\omega) = \frac{1}{\sqrt{q}} \frac{1}{\sqrt{\pi}} \frac{\Gamma(\frac{q}{2})}{\Gamma(\frac{q-1}{2})} \left(1 - \frac{\omega^2}{q} \right)^{\frac{q-3}{2}} \rightarrow \frac{1}{\sqrt{2\pi}} e^{-\frac{\omega^2}{2}} \quad (\text{A.2})$$

as $q \rightarrow \infty$, using the fact that $\frac{\Gamma(\frac{q}{2})}{\Gamma(\frac{q-1}{2})} \sim \sqrt{\frac{q}{2}}$. Therefore the probability that two random Gaussian vectors are not ϵ -orthogonal is upper bounded by

$$\begin{aligned} \Pr(|\rho| \geq \epsilon) &= \Pr(|\omega| \geq \sqrt{q}\epsilon) = 2 \int_{\sqrt{q}\epsilon}^{\sqrt{q}} \frac{1}{\sqrt{2\pi}} e^{-\frac{\omega^2}{2}} d\omega \\ &< \sqrt{\frac{2}{\pi}} e^{-\frac{q\epsilon^2}{2}} (\sqrt{q} - \sqrt{q}\epsilon) < \sqrt{\frac{2q}{\pi}} e^{-\frac{q\epsilon^2}{2}}. \end{aligned} \quad (\text{A.3})$$

To estimate the probability that ϵ -orthogonality is satisfied for a set of N independent Gaussian random vectors, let us consider the following quantity

$$\mathcal{P}(\epsilon, N) := \prod_{k=1}^{N-1} [1 - k \Pr(|\rho| \geq \epsilon)]. \quad (\text{A.4})$$

The above estimation has clear meaning. Given one Gaussian random vector \mathbf{X}_1 , the probability that an independently sampled random vector \mathbf{X}_2 which is not ϵ -orthogonal to \mathbf{X}_1 is $\Pr(|\rho| > \epsilon)$. Similarly, given k i.i.d. Gaussian random vectors $\mathbf{X}_1, \dots, \mathbf{X}_k$, the probability that an independently drawn Gaussian random vector \mathbf{X}_{k+1} which is not ϵ -orthogonal to $\mathbf{X}_1, \dots, \mathbf{X}_k$ is upper bounded by $k \Pr(|\rho| > \epsilon)$. Therefore, we have the estimate in Eq. A.4 for N independent random vectors.

Using Eq. A.3, $\mathcal{P}(\epsilon, N)$ can be computed as follows

$$\begin{aligned} \mathcal{P}(\epsilon, N) &> \prod_{k=1}^{N-1} (1 - k \sqrt{\frac{2q}{\pi}} e^{-\frac{\epsilon^2 q}{2}}) \\ &> (1 - N \sqrt{\frac{2q}{\pi}} e^{-\frac{\epsilon^2 q}{2}})^N \sim e^{-N^2 \sqrt{\frac{2q}{\pi}} e^{-\frac{\epsilon^2 q}{2}}}, \end{aligned}$$

for sufficiently large N and q satisfying $N \sqrt{\frac{2q}{\pi}} e^{-\frac{\epsilon^2 q}{2}} < 1$. If we require $\mathcal{P}(\epsilon, N) \geq 1 - \nu$, then the number of pairwise ϵ -orthogonal i.i.d. Gaussian random vectors is bounded from above by

$$\begin{aligned} e^{-N^2 \sqrt{\frac{2q}{\pi}} e^{-\frac{\epsilon^2 q}{2}}} &\geq 1 - \nu \Rightarrow \\ N &\leq \sqrt[4]{\frac{\pi}{2q}} e^{\frac{\epsilon^2 q}{4}} \left[\log \left(\frac{1}{1-\nu} \right) \right]^{\frac{1}{2}} \quad \blacksquare \end{aligned}$$

Corollary 2. Consider a set of n q -dimensional random Gaussian vectors, we have

$$\lambda_G := \mathbb{E}[|\rho_G|] = \sqrt{\frac{2}{\pi q}}. \quad (\text{A.5})$$

Proof. Given the $g(\rho_G)$ in Theorem 1, we have

$$\begin{aligned} \mathbb{E}[|\rho_G|] &= \int_{-1}^1 |\rho| g(\rho) d\rho = \sqrt{\frac{2q}{\pi}} \int_0^1 \rho (1-\rho^2)^{\frac{q-3}{2}} d\rho \\ &= -\sqrt{\frac{2q}{\pi}} \frac{(1-\rho^2)^{\frac{q-1}{2}}}{q-1} \Big|_0^1 = \sqrt{\frac{2}{\pi q}}, \end{aligned}$$

for large q . \blacksquare

A.2 DISCUSSION ON CONJECTURE 1

In this section, we derive the approximations stated in Conjecture 1 and verify them with empirical simulations.

According to the central limit theorem, the sum of independently and identically distributed random variables with finite variance converges weakly to a normal distribution as the number of random variables approaches infinity. Our derivation relies on the generalized central limit theorem proven by Gnedenko and Kolmogorov in 1954 [Gnedenko et al. 1954].

Theorem A 1. (Generalized Central Limit Theorem [Gnedenko et al. 1954]) Suppose X_1, X_2, \dots is a sequence of i.i.d random variables drawn from the distribution with probability density function $f(x)$ with the following asymptotic behaviour

$$f(x) \simeq \begin{cases} c_+ x^{-(\alpha+1)} & \text{for } x \rightarrow \infty \\ c_- |x|^{-(\alpha+1)} & \text{for } x \rightarrow -\infty, \end{cases} \quad (\text{A.6})$$

where $0 < \alpha < 2$, and c_+, c_- are real positive numbers. Define random variable S_n as a superposition of X_1, \dots, X_n

$$S_n = \frac{\sum_{i=1}^n X_i - C_n}{n^{\frac{1}{\alpha}}}, \quad \text{with}$$

$$C_n = \begin{cases} 0 & \text{if } 0 < \alpha < 1 \\ n^2 \Im(\ln(\phi_X(1/n))) & \text{if } \alpha = 1 \\ n\mathbb{E}[X] & \text{if } 1 < \alpha < 2, \end{cases}$$

where ϕ_X is the characteristic function of a random variable X with probability density function $f(x)$, $\mathbb{E}[X]$ is the expectation value of X , \Im denotes the imaginary part of a variable. Then as the number of summands n approaches infinity, the random variables S_n converge in distribution to a unique stable distribution $S(x; \alpha, \beta, \gamma, 0)$, that is

$$S_n \xrightarrow{d} S(\alpha, \beta, \gamma, 0), \quad \text{for } n \rightarrow \infty,$$

where, α characterizes the power-law tail of $f(x)$ as defined above, and parameters β and γ are given as:

$$\beta = \frac{c_+ - c_-}{c_+ + c_-},$$

$$\gamma = \left[\frac{\pi(c_+ + c_-)}{2\alpha \sin(\frac{\pi\alpha}{2})\Gamma(\alpha)} \right]^{\frac{1}{\alpha}}. \quad (\text{A.7})$$

To be self-contained, we give the definition of stable distributions after [Nolan 2003; Mandelbrot 1960].

Definition A 1. A random variable X follows a stable distribution if its characteristic function can be expressed as

$$\phi(t; \alpha, \beta, \gamma, \mu) = e^{i\mu t - |\gamma t|^\alpha (1 - i\beta \operatorname{sgn}(t)\Phi(\alpha, t))}, \quad (\text{A.8})$$

with $\Phi(\alpha, t)$ defined as

$$\Phi(\alpha, t) = \begin{cases} \tan(\frac{\pi\alpha}{2}) & \text{if } \alpha \neq 1 \\ -\frac{2}{\pi} \log|t| & \text{if } \alpha = 1. \end{cases}$$

Then the probability density function $S(x; \alpha, \beta, \gamma, \mu)$ of the random variable X is given by the Fourier transform of its characteristic function

$$S(x; \alpha, \beta, \gamma, \mu) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \phi(t; \alpha, \beta, \gamma, \mu) e^{-ixt} dx.$$

The parameter α satisfying $0 < \alpha \leq 2$ characterizes the power-law asymptotic limit of the stable distribution, $\beta \in [-1, 1]$ measures the skewness, $\gamma > 0$ is the scale parameter, and $\mu \in \mathbb{R}$ is the shift parameter. Note that the

normal distribution is a typical stable distribution. Other examples with analytical expression include the Cauchy distribution and the Lévy distribution. For the later use, we give the analytical form of the Lévy distribution.

Remark A 1. The probability density function of the Lévy distribution is given by

$$f(x; \gamma, \mu) = \sqrt{\frac{\gamma}{2\pi}} \frac{e^{-\frac{\gamma}{2(x-\mu)}}}{(x-\mu)^{\frac{3}{2}}}, \quad x \geq \mu, \quad (\text{A.9})$$

where μ is the shift parameter and γ is the scale parameter. The Lévy distribution is a special case of the stable distribution $S(x; \alpha, \beta, \gamma, \mu)$ with $\alpha = \frac{1}{2}$ and $\beta = 1$. This can be seen from its characteristic function, which can be written as

$$\phi(t; \gamma, \mu) = e^{i\mu t - |\gamma t|^{1/2}(1 - i \operatorname{sgn}(t))}$$

To derive $g(\rho_C)$ for Cauchy random vectors, we first need the distribution function of X^2 given that the random variable X has a Cauchy distribution.

Lemma A 1. Let X be a Cauchy random variable having the probability density function $f_X(x) = \frac{1}{\pi} \frac{\zeta}{x^2 + \zeta^2}$, where $\zeta > 0$ is the scale parameter. Then the squared variable $Y := X^2$ has the pdf:

$$f_Y(y) = \begin{cases} \frac{1}{\pi} \frac{\zeta}{\sqrt{y}(\zeta^2 + y)} & \text{for } y \geq 0, \\ 0 & \text{otherwise.} \end{cases} \quad (\text{A.10})$$

Proof. $f_Y(y)$ can be derived from $f_X(x)$ by a simple variable transformation $y = g(x) = x^2$. In particular, utilizing the symmetry of $f_X(x)$, we have

$$f_Y(y) = 2 \left| \frac{d}{dy} g^{-1}(y) \right| f_X(g^{-1}(y))$$

$$= \frac{1}{\pi} \frac{\zeta}{\sqrt{y}(\zeta^2 + y)}.$$

■

In the following Lemma we derive the probability density function for $z_{\mathbf{X}, \mathbf{Y}}$, which is defined as $z_{\mathbf{X}, \mathbf{Y}} := \frac{1}{q^2} \frac{X_2^2 + \dots + X_q^2}{X_1^2}$.

Lemma A 2. Let X_1, \dots, X_q be a sequence of i.i.d. random variables drawn from $\mathcal{C}(0, 1)$. Then the random variable $Z_q := \frac{1}{q^2} \frac{X_2^2 + \dots + X_q^2}{X_1^2}$ converges in distribution to

$$f(z) = -\frac{1}{\pi^2} \frac{1}{z^{\frac{3}{2}}} \left[e^{\frac{1}{\pi z}} \operatorname{Ei} \left(-\frac{1}{\pi z} \right) \right], \quad (\text{A.11})$$

as $q \rightarrow \infty$, where $\operatorname{Ei}(x)$ denotes the exponential integral.

Proof. The numerator in Z_q can be regarded as a sum of independent random variables with density function $f_{Y:=X^2}(y) = \frac{1}{\pi} \frac{1}{\sqrt{y(1+y)}}$, see Eq. A.10 with $\zeta = 1$. Thus, we can use the generalized central limit theorem to obtain the density function $g(\frac{1}{q^2} \sum_{i=2}^q X_i^2)$ for the numerator, as $q \rightarrow \infty$.

Note that $f_Y(y) \sim \frac{1}{\pi} y^{-\frac{3}{2}}$ as $y \rightarrow +\infty$. From this asymptotic behaviour we can extract that $c_+ = \frac{1}{\pi}$, $c_- = 0$, and $\alpha = \frac{1}{2}$. Moreover, Eq. A.7 with $\beta = 1$ yields $\gamma = \left[\frac{1}{\sin(\frac{\pi}{4}) \Gamma(\frac{1}{2})} \right]^2 = \frac{2}{\pi}$. In summary, $g(\frac{1}{q^2} \sum_{i=2}^q X_i^2)$ converges to a unique stable distribution $S(\alpha = \frac{1}{2}, \beta = 1, \gamma = \frac{2}{\pi}, \mu = 0)$, which is exactly the Lévy distribution shown in Remark A.1. Hence, we have

$$g\left(\frac{1}{q^2} \sum_{i=2}^q X_i^2\right) \xrightarrow{d} S\left(x; \frac{1}{2}, 1, \frac{2}{\pi}, 0\right) = \frac{1}{\pi} \frac{e^{-\frac{1}{\pi x}}}{x^{\frac{3}{2}}},$$

as $q \rightarrow \infty$. (A.12)

Next, we consider the quotient distribution of two random variables in order to derive the pdf of Z_q . To be more specific, let X and Y be independent non-negative random variables with corresponding probability density function $f_X(x)$ and $f_Y(y)$ over the domains $x \geq 0$ and $y \geq 0$, respectively. Then the cumulative distribution function $F_Z(z)$ of $Z := \frac{Y}{X}$ can be computed by

$$\begin{aligned} F_Z(z) &= \Pr\left(\frac{Y}{X} \leq z\right) = \Pr(Y \leq zX) \\ &= \int_0^\infty \left[\int_0^{y=zx} f_Y(y) dy \right] f_X(x) dx. \end{aligned}$$

Differentiating the cumulative distribution function yields

$$f_Z(z) = \frac{d}{dz} F_Z(z) = \int_0^\infty x f_Y(zx) f_X(x) dx.$$

Following the above procedure, we can obtain the pdf for Z_q as $q \rightarrow \infty$ in case the density functions of the numerator and the denominator are given by Eq. A.12 and Eq. A.10, respectively. That yields

$$\begin{aligned} f(z) &= \frac{1}{\pi^2} \int_0^\infty x \frac{e^{-\frac{1}{\pi zx}}}{(zx)^{\frac{3}{2}}} \frac{1}{\sqrt{x(1+x)}} dx \\ &= \frac{1}{\pi^2} \frac{1}{z^{\frac{3}{2}}} \left[-e^{\frac{1}{\pi z}} \text{Ei}\left(-\frac{x+1}{\pi zx}\right) \right] \Big|_{x=0}^\infty \\ &= -\frac{1}{\pi^2} \frac{1}{z^{\frac{3}{2}}} \left[e^{\frac{1}{\pi z}} \text{Ei}\left(-\frac{1}{\pi z}\right) \right]. \end{aligned}$$

■

In the following we discuss why the density function $g(\rho_C)$ can only be approximated by taking the limit as $q \rightarrow \infty$.

Suppose $\mathbf{X} = (X_1, \dots, X_q)$ and $\mathbf{Y} = (Y_1, \dots, Y_q)$ are Gaussian random variables. To derive $g(\rho_{\mathbf{X}, \mathbf{Y}})$ in Lemma 1, [Cai et al. 2012; Muirhead 2009] compute the density function of $\frac{\alpha^\top \mathbf{X}}{\|\mathbf{X}\|}$ instead, where $\alpha^\top \cdot \alpha = 1$, and $\alpha := \frac{\mathbf{Y}}{\|\mathbf{Y}\|}$. In particular, without loss of generality, they assume $\alpha = (1, 0, \dots, 0)$. The justification for this assumption is that the random variable $\mathbf{X}' := \frac{\mathbf{X}}{\|\mathbf{X}\|}$ is uniformly distributed on the $(q-1)$ -dimensional sphere (see Theorem 1.5.6 in [Muirhead 2009]).

In our case, the distributional uniformity of $\frac{\mathbf{X}}{\|\mathbf{X}\|}$ is not superficial, since the density function of \mathbf{X}' doesn't depend on \mathbf{X}' only through the value of $\mathbf{X}'^\top \mathbf{X}'$. To see this, in the following Lemma, we discuss the distribution function of the normalization $\frac{\mathbf{X}}{\|\mathbf{X}\|}$.

Lemma A.3. *Consider a q -dimensional random vector $\mathbf{X} = (X_1, \dots, X_q)$, where X_1, \dots, X_q are independently and identically drawn from a Cauchy distribution $\mathcal{C}(0, 1)$. Then, as $q \rightarrow \infty$, the normalized random vector $\frac{\mathbf{X}}{\|\mathbf{X}\|} = (X'_1, \dots, X'_q)$ has a joint density function, in which the random variables X'_1, \dots, X'_q are all independent from each other.*

Proof. Without loss of generality, we study the pdf of $X'_1 = \frac{X_1}{\sqrt{X_1^2 + \dots + X_q^2}}$. Similar to the proof of Lemma

A.2, the random variable $Z_q := \frac{X_1^2 + \dots + X_q^2}{X_1^2}$ converges weakly to the distribution with pdf given by Eq. A.11 as $q \rightarrow \infty$, which is independent of the other random variables due to the generalized central limit theorem. Hence, X'_1 can be treated as an independent random variable as $q \rightarrow \infty$. In addition, we obtain the pdf of X'_1 given by

$$f_{X'_1}(x'_1) = -\frac{2}{\pi^2 q^2 x_1'^3} \frac{1}{z_1^{\frac{3}{2}}} \left[e^{\frac{1}{\pi z_1}} \text{Ei}\left(-\frac{1}{\pi z_1}\right) \right],$$

(A.13)

where z_1 is defined as $z_1 := \frac{1}{q^2} \left(\frac{1}{x_1'^2} - 1 \right)$. The arguments can be easily generalized to X'_2, \dots, X'_q . ■

The pdf of the joint distribution $f_{\mathbf{X}'}(x'_1, \dots, x'_q)$ can be written as a product of marginals, that is

$$f_{\mathbf{X}'}(x'_1, \dots, x'_q) = \prod_{i=1}^q f_{X'_i}(x'_i),$$

as $q \rightarrow \infty$. The density function of \mathbf{X}' is not invariant under an arbitrary rotation. Thus, it is not uniformly distributed on S^{q-1} .

The above density function of normalized Cauchy random vectors leads to the following Remark.

Remark A 2. The normalized Cauchy random vector $\mathbf{X}' = \frac{\mathbf{X}}{\|\mathbf{X}\|}$ is sparse in the sense that the density function of its elements can be approximated by a δ -function.

Fig. 1 shows the empirical elements distribution of 1000 normalized Cauchy random vectors. This indicates that in sufficiently high-dimensional spaces the density function of the normalized entries converges to a δ -function. To explain this, recall the Laurent expansion of the density function given in Eq. A.13,

$$f_{X'_1}(x'_1) = \frac{2}{\pi q x_1'^2} - \frac{2}{q^3 x_1'^4} + \frac{4\pi}{q^5 x_1'^6} + \mathcal{O}\left(\frac{1}{q^7 x_1'^8}\right). \quad (\text{A.14})$$

This expansion converges to zero almost everywhere except for $x'_1 = 0$ as $q \rightarrow \infty$.

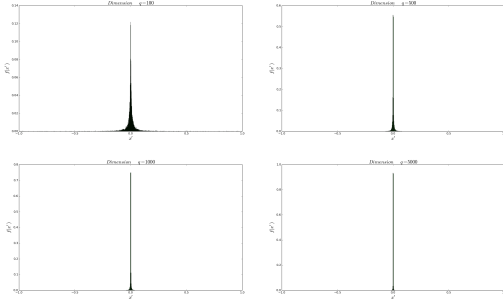


Figure 1: Empirical distributions of 10000 normalized Cauchy random vectors with dimensions $q = 100, 500, 1000, 5000$.

In the following, we provide a full derivation of $g(\rho_C)$ proposed in the Conjecture 1.

Conjecture 1. Let $\mathbf{X}_1, \dots, \mathbf{X}_n$ be independent q -dimensional random vectors whose elements are independently and identically drawn from a Cauchy distribution $\mathcal{C}(0, 1)$. Let Θ_{ij} be the angle between \mathbf{X}_i and \mathbf{X}_j . Then, as $q \rightarrow \infty$, $\rho_{ij} := \cos \Theta_{ij} \in [-1, 1]$, $1 \leq i < j \leq n$ are pairwise i.i.d. with density function approximated by

$$g(\rho_C) = -\frac{2}{\pi^2 q^2 \rho_C^3} \cdot \frac{1}{z^{\frac{3}{2}}} \left[e^{\frac{1}{\pi z}} \text{Ei}\left(-\frac{1}{\pi z}\right) \right], \quad (\text{A.15})$$

where $z := \frac{1}{q^2} \left(\frac{1}{\rho_C^2} - 1 \right)$.

Given two Cauchy random vectors $\mathbf{X} = (X_1, \dots, X_q)$ and $\mathbf{Y} = (Y_1, \dots, Y_q)$, $\rho_{\mathbf{X}, \mathbf{Y}}$ is approximated by $\rho_{\mathbf{X}, \mathbf{Y}} \approx \frac{X_1 Y_1}{\sqrt{X_1^2 + \dots + X_q^2}}$.

Furthermore, we introduce the new variable $z_{\mathbf{X}, \mathbf{Y}} := \frac{1}{q^2} \left(\frac{1}{\rho_{\mathbf{X}, \mathbf{Y}}} - 1 \right)$. From Lemma A 2 we have the density function $\hat{g}(z_{\mathbf{X}, \mathbf{Y}})$. Then, $g(\rho_{\mathbf{X}, \mathbf{Y}})$ can be directly obtained from $\hat{g}(z_{\mathbf{X}, \mathbf{Y}})$ by a variable transform, that is

$g(\rho_{\mathbf{X}, \mathbf{Y}}) = \left| \frac{dz}{d\rho} \right| \hat{g}(z_{\mathbf{X}, \mathbf{Y}})$. With $\left| \frac{dz}{d\rho} \right| = \frac{2}{q^2 \rho^3}$ we immediately get Eq. A.15 as the density function for $\rho_{\mathbf{X}, \mathbf{Y}}$.

Assume that Eq. A.15 is valid as $q \rightarrow \infty$. In the following we show that $\{\rho_{ij} | 1 \leq i < j \leq n\}$ are i.i.d random variables. First, notice that ρ_{ij} and ρ_{kl} are independent if $\{i, j\} \cap \{k, l\} = \emptyset$. It is left to prove that $\rho_{\mathbf{X}, \mathbf{Y}}$ and $\rho_{\mathbf{X}, \mathbf{Z}}$ are independent, given that $\mathbf{X}, \mathbf{Y}, \mathbf{Z}$ are independent random variables.

To prove the independence, consider $\mathbb{E}[h_1(\rho_{\mathbf{X}, \mathbf{Y}}) h_2(\rho_{\mathbf{X}, \mathbf{Z}})]$, where h_1 and h_2 are arbitrary bounded functions. Since \mathbf{X}, \mathbf{Y} , and \mathbf{Z} are independent,

$$\begin{aligned} \mathbb{E}[h_1(\rho_{\mathbf{X}, \mathbf{Y}}) \cdot h_2(\rho_{\mathbf{X}, \mathbf{Z}})] &= \mathbb{E}[\mathbb{E}[h_1(\rho_{\mathbf{X}, \mathbf{Y}}) \cdot h_2(\rho_{\mathbf{X}, \mathbf{Z}}) | \mathbf{X}]] \\ &= \mathbb{E}[\mathbb{E}[h_1(\rho_{\mathbf{X}, \mathbf{Y}}) | \mathbf{X}] \cdot \mathbb{E}[h_2(\rho_{\mathbf{X}, \mathbf{Z}}) | \mathbf{X}]] \end{aligned}$$

Given \mathbf{X} , the probability density function of $\rho_{\mathbf{X}, \mathbf{Y}}$ is independent of \mathbf{X} . Thus, $\mathbb{E}[h_1(\rho_{\mathbf{X}, \mathbf{Y}}) | \mathbf{X}] = \int_{-1}^1 h_1(\rho_{\mathbf{X}, \mathbf{Y}}) g(\rho_{\mathbf{X}, \mathbf{Y}}) d\rho = \mathbb{E}[h_1(\rho_{\mathbf{X}, \mathbf{Y}})]$, and similarly $\mathbb{E}[h_2(\rho_{\mathbf{X}, \mathbf{Z}}) | \mathbf{X}] = \mathbb{E}[h_2(\rho_{\mathbf{X}, \mathbf{Z}})]$. It gives,

$$\mathbb{E}[h_1(\rho_{\mathbf{X}, \mathbf{Y}}) \cdot h_2(\rho_{\mathbf{X}, \mathbf{Z}})] = \mathbb{E}[h_1(\rho_{\mathbf{X}, \mathbf{Y}})] \cdot \mathbb{E}[h_2(\rho_{\mathbf{X}, \mathbf{Z}})],$$

This concludes that $\rho_{\mathbf{X}, \mathbf{Y}}$ and $\rho_{\mathbf{X}, \mathbf{Z}}$ are also independent. ■

Recall that the derivation of Eq. A.15 uses the generalized central limit theorem which requires the limiting condition $q \rightarrow \infty$. Therefore it is important to check how the dimensionality q effects the quality of the prediction.

Fig. 2 displays the empirical distribution of ρ , that is $g(\rho) = \sum_{1 \leq i < j \leq n} \delta_{\rho_{ij}}$, and the theoretical prediction in Eq. A.15 for various dimensions q . For the simulation, $n = 10000$ random vectors are drawn independently from $\mathcal{C}(0, 1)$. We use the leading orders of the Laurent series of Eq. A.15 to represent the theoretical predictions.

It can be seen that for a sufficiently high-dimensional space, say $q = 2000$, the theoretical prediction fits the simulation very well. Moreover, the pairwise angles among Cauchy random vectors converge to $\frac{\pi}{2}$ as the dimensionality increases.

It implies that in high-dimensional spaces the distributional uniformity of normalized Cauchy random vectors could be tenable. We explain this in an intuitive way. According to Remark A 2, each element in the normalized variable converges independently in distribution to a Dirac δ -function, which can be constructed as the limit of a sequence of zero-centered normal distribution

$$f_{X'_i}(x'_i) = \frac{1}{a\sqrt{\pi}} e^{-\frac{x_i'^2}{a^2}} \quad \text{for } a \rightarrow 0^+.$$

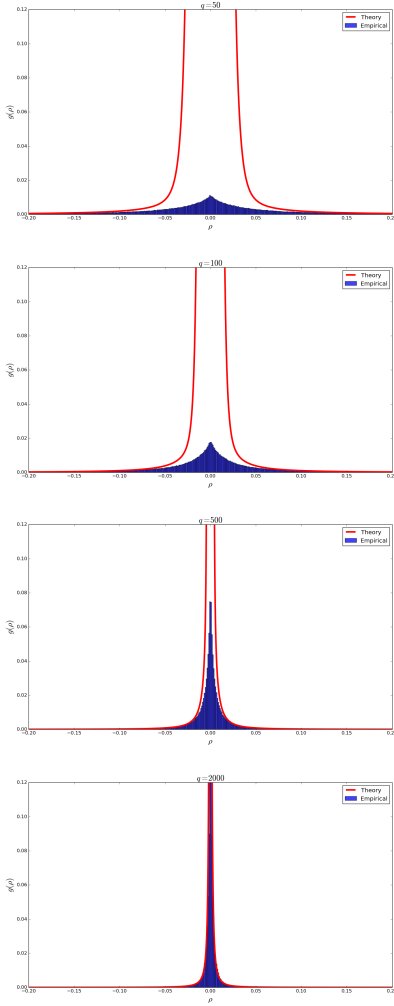


Figure 2: Comparisons between empirical distributions and theoretical predictions of ρ_C for various dimensions, $q = 50, 100, 500, 2000$.

Thus, following Lemma A 3, the density function of $f_{\mathbf{X}'}(x'_1, \dots, x'_q)$ can be approximated by

$$f_{\mathbf{X}'}(x'_1, \dots, x'_q) = \left(\frac{1}{a\sqrt{\pi}} \right)^q e^{-\frac{\mathbf{x}'^T \mathbf{x}'}{a^2}} \quad \text{for } a \rightarrow 0^+.$$

This joint distribution is invariant under an arbitrary orthogonal rotation. Thus, it is a spherical distribution, as well as a uniform distribution on S^{q-1} . A rigorous proof of this result is still necessary. However, it is beyond the scope of this work.

A.3 DERIVATION OF COROLLARY 3

Corollary 3. *Consider a set of independent q -dimensional Cauchy random vectors which are pairwise*

ϵ -orthogonal with probability $1 - \nu$. Then the number of such Cauchy random vectors is bounded by

$$N \leq \sqrt{\frac{\pi \epsilon q}{4}} \left[\log \left(\frac{1}{1 - \nu} \right) \right]^{\frac{1}{2}}. \quad (\text{A.16})$$

Proof. The derivation of this bound is similar to that of Corollary 2. The probability, that two random vectors whose elements are independently and identically Cauchy distributed are not ϵ -orthogonal, is bounded from above by

$$\Pr(|\rho| \geq \epsilon) = 2 \int_{\epsilon}^1 \frac{2}{\pi q \rho^2} d\rho < \frac{4}{\pi q} \frac{1}{\epsilon},$$

where only the leading order Laurent expansion of Eq. A.15 is considered. Then the quantity $\mathcal{P}(\epsilon, N)$ can be estimated as follows,

$$\begin{aligned} \mathcal{P}(\epsilon, N) &:= \prod_{k=1}^{N-1} [1 - k \Pr(|\rho| \geq \epsilon)] > \prod_{k=1}^{N-1} \left(1 - k \frac{4}{\pi \epsilon q} \right) \\ &> \left(1 - N \frac{4}{\pi \epsilon q} \right)^N \sim e^{-N^2 \frac{4}{\pi \epsilon q}}, \end{aligned}$$

for sufficiently large N , and $q \rightarrow \infty$, with $N \frac{4}{\pi \epsilon q} < 1$. If we require $\mathcal{P}(\epsilon, N) \geq 1 - \nu$, then the number of pairwise ϵ -orthogonal i.i.d. Cauchy random vectors is upper bounded by

$$e^{-N^2 \frac{4}{\pi \epsilon q}} \geq 1 - \nu \Rightarrow N \leq \sqrt{\frac{\pi \epsilon q}{4}} \left[\log \left(\frac{1}{1 - \nu} \right) \right]^{\frac{1}{2}}$$

■

A.4 BINDING WITH CORRELATION OR CONVOLUTION

The filtered mean rank scores with different binding operations are compared in Fig. 3.

Now we give a heuristic explanation. For the sake of simplicity, consider only one semantic triple (s, p, o) . For the binding with circular correlation the holistic representations are given by $\mathbf{h}_s^{\text{corr}} = \mathbf{r}_p \star \mathbf{r}_o + \xi \mathbf{r}_s$, $\mathbf{h}_p^{\text{corr}} = \mathbf{r}_s \star \mathbf{r}_o + \xi \mathbf{r}_p$, and $\mathbf{h}_o^{\text{corr}} = \mathbf{r}_p \star \mathbf{r}_s + \xi \mathbf{r}_o$.

On the other hand, for the binding with convolution, the holistic representations given by: $\mathbf{h}_s^{\text{conv}} = \mathbf{r}_p * \mathbf{r}_o + \xi \mathbf{r}_s$, $\mathbf{h}_p^{\text{conv}} = \mathbf{r}_s * \mathbf{r}_o + \xi \mathbf{r}_p$, and $\mathbf{h}_o^{\text{conv}} = \mathbf{r}_p * \mathbf{r}_s + \xi \mathbf{r}_o$.

Suppose that the subject needs to be retrieved and recalled using holistic representations only. To quantify the retrieval quality, a similarity $s^{\text{corr/conv}}$ is introduced for different binding operators. In particular, for binding with circular correlation $s^{\text{corr}} := \mathbf{h}_s^{\text{corr}} \top (\mathbf{h}_p^{\text{corr}} * \mathbf{h}_o^{\text{corr}})$,

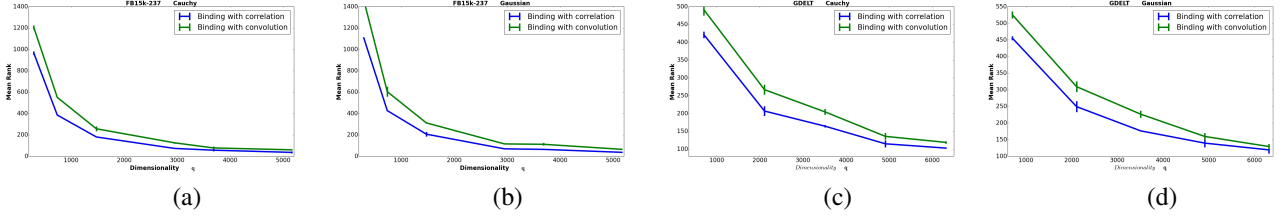


Figure 3: Comparison of the filtered MR scores for binding with convolution and binding with correlation (a) for FB15k-237 with Cauchy initialization, (b) for FB15k-237 with Gaussian initialization, (c) for GDELT dataset with Cauchy initialization, (d) for GDELT with Gaussian initialization

while for binding with circular convolution $s^{\text{conv}} := \mathbf{h}_s^{\text{conv}} \top (\mathbf{h}_p^{\text{conv}} \star \mathbf{h}_o^{\text{conv}})$.

Before any further derivations, recall that circular correlation can be computed in log-linear complexity via

$$\mathbf{a} \star \mathbf{b} = \mathcal{F}^{-1} \left(\overline{\mathcal{F}(\mathbf{a})} \odot \mathcal{F}(\mathbf{b}) \right),$$

where $\mathcal{F}(\cdot)$ denotes the *fast Fourier transform* and $\mathcal{F}^{-1}(\cdot)$ its inverse, and the bar denotes the complex conjugate of a complex-valued vector. Moreover, circular convolution can also be computed via *fast Fourier transforms*

$$\mathbf{a} \star \mathbf{b} = \mathcal{F}^{-1} \left(\mathcal{F}(\mathbf{a}) \odot \mathcal{F}(\mathbf{b}) \right).$$

First we compute the similarity s^{corr}

$$\begin{aligned} s^{\text{corr}} &= \mathbf{h}_s^{\text{corr}} \top (\mathbf{h}_p^{\text{corr}} \star \mathbf{h}_o^{\text{corr}}) \\ &= (\mathbf{r}_p \star \mathbf{r}_o + \xi \mathbf{r}_s) \top [(\mathbf{r}_s \star \mathbf{r}_o + \xi \mathbf{r}_p) \star (\mathbf{r}_p \star \mathbf{r}_s + \xi \mathbf{r}_o)] \\ &= (\mathbf{r}_p \star \mathbf{r}_o + \xi \mathbf{r}_s) \top \underbrace{[(\mathbf{r}_s \star \mathbf{r}_o) \star (\mathbf{r}_p \star \mathbf{r}_s)]}_{\textcircled{1}} + \\ &\quad \underbrace{\xi (\mathbf{r}_s \star \mathbf{r}_o) \star \mathbf{r}_o}_{\textcircled{2}} + \underbrace{\xi \mathbf{r}_p \star (\mathbf{r}_p \star \mathbf{r}_s)}_{\textcircled{3}} + \xi^2 \mathbf{r}_p \star \mathbf{r}_o. \end{aligned}$$

Using that

$$\begin{aligned} \textcircled{1} &= \mathcal{F}^{-1} \left[\overline{\mathcal{F}(\mathbf{r}_s)} \odot \mathcal{F}(\mathbf{r}_o) \odot \overline{\mathcal{F}(\mathbf{r}_p)} \odot \mathcal{F}(\mathbf{r}_s) \right] \approx \mathbf{r}_p \star \mathbf{r}_o, \\ \textcircled{2} &= \mathcal{F}^{-1} \left[\overline{\mathcal{F}(\mathbf{r}_s)} \odot \mathcal{F}(\mathbf{r}_o) \odot \mathcal{F}(\mathbf{r}_o) \right] = \text{Noise}, \\ \textcircled{3} &= \mathcal{F}^{-1} \left[\mathcal{F}(\mathbf{r}_p) \odot \overline{\mathcal{F}(\mathbf{r}_p)} \odot \mathcal{F}(\mathbf{r}_s) \right] \approx \mathbf{r}_s, \end{aligned}$$

yields

$$\begin{aligned} s^{\text{corr}} &\approx (\mathbf{r}_p \star \mathbf{r}_o + \xi \mathbf{r}_s) \top [\mathbf{r}_p \star \mathbf{r}_o + \xi \mathbf{r}_s + \text{Noise}] \\ &\approx (1 + \xi^2) + \text{Noise}. \end{aligned}$$

The similarity s^{conv} can be computed in a similar way,

$$\begin{aligned} s^{\text{conv}} &= \mathbf{h}_s^{\text{conv}} \top (\mathbf{h}_p^{\text{conv}} \star \mathbf{h}_o^{\text{conv}}) \\ &= (\mathbf{r}_p \star \mathbf{r}_o + \xi \mathbf{r}_s) \top [(\mathbf{r}_s \star \mathbf{r}_o + \xi \mathbf{r}_p) \star (\mathbf{r}_p \star \mathbf{r}_s + \xi \mathbf{r}_o)] \\ &= (\mathbf{r}_p \star \mathbf{r}_o + \xi \mathbf{r}_s) \top \underbrace{[(\mathbf{r}_s \star \mathbf{r}_o) \star (\mathbf{r}_p \star \mathbf{r}_s)]}_{\textcircled{1}} + \\ &\quad \underbrace{\xi (\mathbf{r}_s \star \mathbf{r}_o) \star \mathbf{r}_o}_{\textcircled{2}} + \underbrace{\xi \mathbf{r}_p \star (\mathbf{r}_p \star \mathbf{r}_s)}_{\textcircled{3}} + \xi^2 \mathbf{r}_p \star \mathbf{r}_o. \end{aligned}$$

Moreover, using that

$$\begin{aligned} \textcircled{1} &= \mathcal{F}^{-1} \left[\overline{\mathcal{F}(\mathbf{r}_s)} \odot \overline{\mathcal{F}(\mathbf{r}_o)} \odot \mathcal{F}(\mathbf{r}_p) \odot \mathcal{F}(\mathbf{r}_s) \right] \approx \mathbf{r}_o \star \mathbf{r}_p, \\ \textcircled{2} &= \mathcal{F}^{-1} \left[\overline{\mathcal{F}(\mathbf{r}_s)} \odot \overline{\mathcal{F}(\mathbf{r}_o)} \odot \mathcal{F}(\mathbf{r}_o) \right] \approx \mathbf{r}_s, \\ \textcircled{3} &= \mathcal{F}^{-1} \left[\overline{\mathcal{F}(\mathbf{r}_p)} \odot \mathcal{F}(\mathbf{r}_p) \odot \mathcal{F}(\mathbf{r}_s) \right] \approx \mathbf{r}_s, \end{aligned}$$

leads to

$$\begin{aligned} s^{\text{conv}} &\approx (\mathbf{r}_p \star \mathbf{r}_o + \xi \mathbf{r}_s) \top [\mathbf{r}_o \star \mathbf{r}_p + 2\xi \mathbf{r}_s + \text{Noise}] \\ &\approx 2\xi^2 + \text{Noise}. \end{aligned}$$

The optimal hyper-parameter requires $\xi < 1$ which in turn yields $s^{\text{corr}} > s^{\text{conv}}$. From the derivation of s^{corr} , we have that the subject-object association pair stored in $\mathbf{h}_p^{\text{corr}}$ contributes the most in $s^{\text{corr}} \approx 1 + \xi^2$ via the term $\textcircled{1}$.

A.5 APPROXIMATION OF $\rho_{\mathbf{r}'_o, \mathbf{h}_o}$

Here we provide a heuristic study on the relations between hyper-parameter ξ , $\lambda_{G/C}$, and the average number of association pairs N_a . Recall that ξ was introduced for holistic representations, and $\lambda_{G/C}$ is defined as $\lambda_{G/C} := \mathbb{E}[|\rho_{G/C}|]$.

Consider a subject s . The predicate-object pair (p, o) is stored in the holistic representation \mathbf{h}_s along with the other $N_a - 1$ pairs. This means

$$\mathbf{h}_s = \xi N_a \mathbf{r}_s + \mathbf{r}_p \star \mathbf{r}_o + \sum_{i=2}^{N_a} \mathbf{r}_{p_i} \star \mathbf{r}_{o_i}.$$

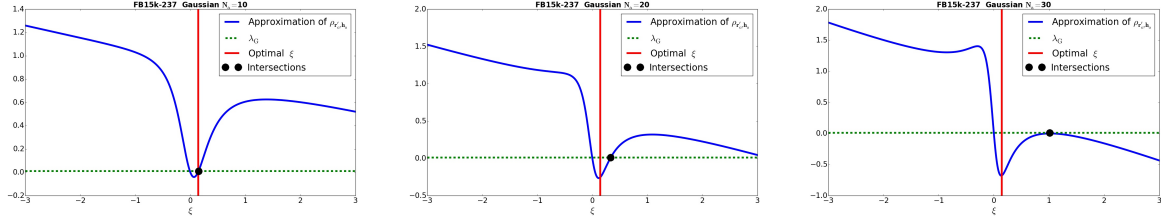


Figure 4: Approximations of $\rho_{\mathbf{r}'_o, \mathbf{h}_o}(\xi, N_a)$ in the case of Gaussian holistic representations with (a): $N_a = 10$ (b): $N_a = 20$ (c): $N_a = 30$. We use the experiment setting with dimensionality $q = 5200$, $\lambda_G = 0.0111$, and optimal $\xi = 0.14$.

Suppose that we aim to identify the object in the triple (s, p, \cdot) via \mathbf{h}_s and \mathbf{h}_p , where \mathbf{h}_p is the holistic representation for the predicate p . We further assume that up to N_a subject-object pairs can be stored in \mathbf{h}_p having high enough fidelity, then

$$\mathbf{h}_p = \xi N_a \mathbf{r}_p + \sum_{k=1}^{N_a} \mathbf{r}_{s_k} \star \mathbf{r}_{o_k}.$$

To retrieve the object o , the decoding via circular convolution is obtained as follows

$$\begin{aligned} \mathbf{r}'_o &= \mathbf{h}_p \star \mathbf{h}_s \\ &\approx \xi N_a \mathbf{r}_o + \xi^2 N_a^2 (\mathbf{r}_p \star \mathbf{r}_s) + \xi N_a \sum_{i=2}^{N_a} [\mathbf{r}_p \star (\mathbf{r}_{p_i} \star \mathbf{r}_{o_i})] \\ &\quad + \xi N_a \sum_{k=1}^{N_a} [(\mathbf{r}_{s_k} \star \mathbf{r}_{o_k}) \star \mathbf{r}_s] + \sum_{k=1}^{N_a} [(\mathbf{r}_{s_k} \star \mathbf{r}_{o_k}) \star (\mathbf{r}_p \star \mathbf{r}_o)] \\ &\quad + \sum_{k=1, i=2}^{N_a, N_a} [(\mathbf{r}_{s_k} \star \mathbf{r}_{o_k}) \star (\mathbf{r}_{p_i} \star \mathbf{r}_{o_i})] \\ &= \xi N_a \mathbf{r}_o + \xi^2 N_a^2 \mathbf{b}_1 + \xi N_a \sum_{i=2}^{N_a} \mathbf{b}_i + \xi N_a \sum_{k=1}^{N_a} \mathbf{c}_k \\ &\quad + \sum_{k=1}^{N_a} \mathbf{d}_k + \sum_{k=1, i=2}^{N_a, N_a} \mathbf{e}_{ki}, \end{aligned}$$

where \mathbf{b}_i , \mathbf{c}_k , \mathbf{d}_k , and \mathbf{e}_{ki} with $i, k = 1, \dots, N_a$ are approximately normalized Gaussian/Cauchy random vectors. This is due to the fact that in high-dimensional spaces both circular correlation and circular convolution of two normalized Gaussian/Cauchy random vectors is approximately a normalized Gaussian/Cauchy random vectors.

After decoding with circular convolutions, the decoded noisy version of the object needs to be recalled with \mathbf{h}_o which is the holistic representation of o . As before, N_a predicate-subject association pairs are assumed to be

stored in the holistic representation of o , with

$$\mathbf{h}_o = \xi N_a \mathbf{r}_o + \sum_{j=1}^{N_a} \mathbf{r}_{p_j} \star \mathbf{r}_{s_j} = \xi N_a \mathbf{r}_o + \sum_{j=1}^{N_a} \mathbf{f}_j,$$

where \mathbf{f}_j , $j = 1, \dots, N_a$ are approximately normalized Gaussian/Cauchy random vectors.

In order to recall the object successfully, the angle between \mathbf{r}'_o and \mathbf{h}_o should be smaller than the expected absolute angle between two arbitrary vectors, namely $\theta_{\mathbf{r}'_o, \mathbf{h}_o} < \mathbb{E}[\theta_{G/C}]$. Given the definition of λ , equivalently, it requires $\rho_{\mathbf{r}'_o, \mathbf{h}_o} > \lambda_{G/C}$.

Now we turn to approximate the numerator of $\rho_{\mathbf{r}'_o, \mathbf{h}_o}$, that is $\mathbf{r}'_o{}^T \mathbf{h}_o$. Recall that, in general, the expectation of the dot product of two normalized, independent random vectors equals 0 due to the symmetry of the density function $g(\rho_{G/C})$. Therefore, in the following approximation we only consider noisy terms which are directly related to \mathbf{r}_o as adverse effects to a successful retrieval and treat other terms as white noisy with zero expectation. This yields,

$$\begin{aligned} \mathbf{r}'_o{}^T \mathbf{h}_o &\approx \xi^2 N_a^2 + \xi N_a \sum_{j=1}^{N_a} (\mathbf{r}_o^T \mathbf{f}_j) + \xi^3 N_a^3 (\mathbf{r}_o^T \mathbf{b}_1) + \xi^2 N_a^2 \sum_{i=2}^{N_a} (\mathbf{r}_o^T \mathbf{b}_i) \\ &\quad + \xi^2 N_a^2 \sum_{k=1}^{N_a} (\mathbf{r}_o^T \mathbf{c}_k) + \xi N_a \sum_{k=1}^{N_a} (\mathbf{r}_o^T \mathbf{d}_k) + \xi N_a \sum_{k=1, i=2}^{N_a, N_a} (\mathbf{r}_o^T \mathbf{e}_{ki}) \\ &> \xi^2 N_a^2 - (\xi N_a^2 + \xi^3 N_a^3 + \xi^2 N_a^2 (N_a - 1) + \xi^2 N_a^3 \\ &\quad + \xi N_a^2 + \xi N_a^2 (N_a - 1)) \lambda_{G/C} \\ &= \xi^2 N_a^2 - (\xi^3 N_a^3 + 2\xi^2 N_a^3 - \xi^2 N_a^2 + \xi N_a^2 + \xi N_a^3) \lambda_{G/C}. \end{aligned}$$

Furthermore, the denominator of $\rho_{\mathbf{r}'_o, \mathbf{h}_o}$ can be approximated in the same way. More concretely, we have

$$\begin{aligned} \|\mathbf{r}'_o\| \cdot \|\mathbf{h}_o\| &< \xi^2 N_a^2 + N_a + 2\xi N_a^2 \lambda_{G/C} \\ &\quad + N_a(N_a - 1) \lambda_{G/C}. \end{aligned}$$

Combining these results, a sufficient condition to retrieve

the object correctly is given by

$$\begin{aligned} \rho_{\mathbf{r}'_o, \mathbf{h}_o} &> \\ \frac{\xi^2 N_a^2 - (\xi^3 N_a^3 + 2\xi^2 N_a^3 - \xi^2 N_a^2 + \xi N_a^2 + \xi N_a^3) \lambda_{G/C}}{\xi^2 N_a^2 + N_a + 2\xi N_a^2 \lambda_{G/C} + N_a(N_a - 1) \lambda_{G/C}} \\ &> \lambda_{G/C}. \end{aligned} \quad (\text{A.17})$$

Consider the experimental setting for the memorization task on the FB15k-237 dataset: The dimensionality of the holistic representations is $q = 5200$, $\lambda_G(q = 5200) = 0.0111$, and $\lambda_C(q = 5200) = 0.00204$. Fig. 4 displays the above approximation of $\rho_{\mathbf{r}'_o, \mathbf{h}_o}(\xi, N_a)$ for Gaussian initializations.

After performing grid search, the optimal ξ is found to be close to the intersection of the curve $\rho_{\mathbf{r}'_o, \mathbf{h}_o}(\xi, N_a = 10)$ and the threshold λ_G . However, for $N_a > 30$, no intersection points on $\xi > 0$ exists. This explains why Gaussian holistic representations have lower memory capacity compared to Cauchy holistic representations.

More comparisons between Gaussian and Cauchy initializations can be found in Fig. 5.

A.6 HOLISTIC ENCODING ALGORITHM

Algorithm 1 Holistic Encoding

Require: hyper-parameter ξ

- 1: **for** $i = 1, \dots, N_e$ **do**
- 2: Draw $\tilde{\mathbf{r}}_{e_i}^{G/C}$ from Gaussian or Cauchy
- 3: $\mathbf{r}_{e_i}^{G/C} \leftarrow \text{Norm}(\tilde{\mathbf{r}}_{e_i}^{G/C})$
- 4: **for** $i = 1, \dots, N_p$ **do**
- 5: Draw $\tilde{\mathbf{r}}_{p_i}^{G/C}$ from Gaussian or Cauchy
- 6: $\mathbf{r}_{p_i}^{G/C} \leftarrow \text{Norm}(\tilde{\mathbf{r}}_{p_i}^{G/C})$
- 7: **for** $i = 1, \dots, N_e$ **do**
- 8: Extract $\mathcal{S}^s(e_i), \mathcal{S}^o(e_i)$ from Database
- 9: $\mathbf{h}_{e_i}^s \leftarrow \sum_{(p,o) \in \mathcal{S}^s(e_i)} [\text{Norm}(\mathbf{r}_p \star \mathbf{r}_o) + \xi \mathbf{r}_{e_i}]$
- 10: $\mathbf{h}_{e_i}^o \leftarrow \sum_{(s,p) \in \mathcal{S}^o(e_i)} [\text{Norm}(\mathbf{r}_p \star \mathbf{r}_s) + \xi \mathbf{r}_{e_i}]$
- 11: $\mathbf{h}_{e_i} \leftarrow \mathbf{h}_{e_i}^s + \mathbf{h}_{e_i}^o$
- 12: **for** $i = 1, \dots, N_p$ **do**
- 13: Extract $\mathcal{S}(p_i)$ from Database
- 14: $\mathbf{h}_{p_i} \leftarrow \sum_{(s,o) \in \mathcal{S}(p_i)} [\text{Norm}(\mathbf{r}_s \star \mathbf{r}_o) + \xi \mathbf{r}_{p_i}]$

Remark:

Normalizing initial random vectors can assist the analysis of memory capacities via different sampling schemes. For example, for the derivation of retrieval condition Eq. A.17 we heavily rely on the fact that the dot product of two random vectors - say $\mathbf{r}_i \cdot \mathbf{r}_j$, where \mathbf{r}_i and \mathbf{r}_j are

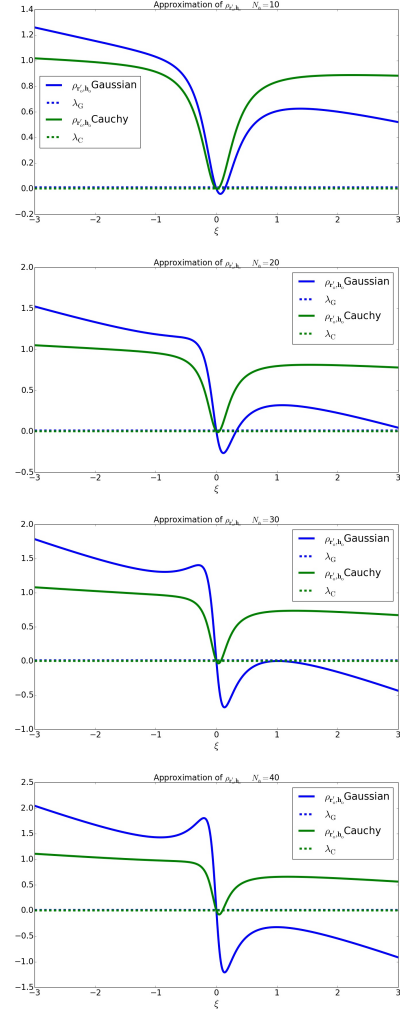


Figure 5: Comparison of $\rho_{\mathbf{r}'_o, \mathbf{h}_o}(\xi, N_a)$ for Gaussian (blue) and Cauchy (green) holistic representations with (a): $N_a = 10$ (b): $N_a = 20$ (c): $N_a = 30$ (d): $N_a = 40$.

randomly sampled and normalized - is just ρ_{ij} . In the memorization task, since triples are recalled by comparing the angles (a.k.a cosine similarity) between decoded noisy vector and all other holistic vectors, normalization does not effect the recall scores.

A.7 NOTATIONS

In Table 1 and Table 2, we summary important notations introduced in Section 3 and 4, respectively.

A.8 FURTHER EXPERIMENTAL DETAILS

After searching for the optimal hyper-parameter ξ for holistic encoding, holistic representations with superior

Table 1: Notations for ϵ -orthogonality

Symbol	Meaning
\mathbf{X}	q -dimensional random variable with elements drawn from Gaussian or Cauchy distribution
Θ_{ij}	Angle between two random variables \mathbf{X}_i and \mathbf{X}_j
ρ_{ij}	Cosine of the angle between random variables \mathbf{X}_i and \mathbf{X}_j
$g(\rho_G)$	Asymptotic density function of ρ_{ij} given an ensemble of Gaussian random variables \mathbf{X}_i , $i = 1, \dots, n$, with $n \rightarrow \infty$
$g(\rho_C)$	Asymptotic density function of ρ_{ij} given an ensemble of Cauchy random variables \mathbf{X}_i , $i = 1, \dots, n$, with $n \rightarrow \infty$
λ_G	Expectation value of $ \rho_G $
λ_C	Expectation value of $ \rho_C $

memory capacity will be fixed and applied to the next inference tasks.

The architecture is a simple 2-layered fully-connected neural network, which map high-dimensional holistic representations ($q = 3600$) of subjects, predicates, and objects to low-dimensional ($h_2 = 256$) representations, separately. We choose ReLU as the activation function for faster training, and batch normalization after the hidden-layer for regularization. In order to reduce the number of trainable parameters, the network has a bottleneck structure with the dimensionality of the hidden-layer $h_1 = 64$. The extracted low-dimensional features are then combined via tri-linear dot-product, similar to DISTMULT.

In summary, given a triple (s, p, o) the scoring function η_{spo} takes the following form:

$$\eta_{spo} = \langle \text{BN}(\text{ReLU}(\mathbf{h}_s \mathbf{W}_1^e)) \mathbf{W}_2^e, \\ \text{BN}(\text{ReLU}(\mathbf{h}_p \mathbf{W}_1^p)) \mathbf{W}_2^p, \\ \text{BN}(\text{ReLU}(\mathbf{h}_o \mathbf{W}_1^e)) \mathbf{W}_2^e \rangle,$$

where $\mathbf{h}_s, \mathbf{h}_o$ are the holistic representations for the subject s and object o ; \mathbf{h}_p is the holistic representation for the predicate p . Note that there are two separate networks for extracting low-dimensional features of entities and predicates, respectively. In particular, $\mathbf{W}_1^e \in \mathbb{R}^{q \times h_1}$ and $\mathbf{W}_2^e \in \mathbb{R}^{h_1 \times h_2}$ are shared weights for entities, including subjects and objects; $\mathbf{W}_1^p \in \mathbb{R}^{q \times h_1}$ and $\mathbf{W}_2^p \in \mathbb{R}^{h_1 \times h_2}$ are shared weights for predicates.

For training the model, we minimize the following binary

Table 2: Notations for holistic representations

Symbol	Meaning
$*$	Circular convolution
\star	Circular correlation
Norm	Normalization operator, $\text{Norm}(\mathbf{r}) := \frac{\mathbf{r}}{\ \mathbf{r}\ }$
N_e	Number of entities in the KG
N_p	Number of predicates in the KG
N_a	Average number of association pairs encoded in holistic representations of entities
$\mathbf{r}_{e_i}^{G/C}$	Random initialization of entity e_i with elements drawn from Gaussian or Cauchy distribution
$\mathbf{r}_{p_i}^{G/C}$	Random initialization of predicate p_i with elements drawn from Gaussian or Cauchy distribution
$\mathbf{h}_{e_i}^s$	Holistic representation of entity e_i as subject
$\mathbf{h}_{e_i}^o$	Holistic representation of entity e_i as object
\mathbf{h}_{e_i}	Overall holistic representation of entity e_i
\mathbf{h}_{p_i}	Holistic representation of predicate p_i
ξ	Hyper-parameter for holistic encoding

cross-entropy loss with l_2 regularization:

$$\mathcal{L} = -\frac{1}{m} \sum_{i=1}^m (y_i \cdot \log(\sigma(\eta_{x_i})) + (1 - y_i) \cdot \log(1 - \sigma(\eta_{x_i}))) + \lambda \|\mathcal{A}\|_2^2,$$

where the label vector y_i has dimension $\{0, 1\}^{1 \times N}$ for 1-N scoring to accelerate the link prediction tasks. To be more specific, during the training given a triple (s, p, o) , we take the subject-predicate pair (s, p) and rank it against all object entities $o \in \mathcal{E}$; take the predicate-object pair (p, o) and rank it against all subject entities $s \in \mathcal{E}$ simultaneously as well.

Hyper-parameters in the HOLNN_G and HOLNN_C are optimized via grid search with respect to the mean reciprocal rank (MRR). The ranges for grid search are as follows - learning rate $\{0.001, 0.003, 0.005\}$, l_2 regularization parameter $\{0., 0.01, 0.05\}$, decay parameter in the batch normalization $\{0.99, 0.9, 0.8, 0.7\}$, and batch size $\{1000, 3000, 5000\}$.

References

- Cai, T Tony and Tiefeng Jiang (2012). “Phase transition in limiting distributions of coherence of high-dimensional random matrices”. In: *Journal of Multivariate Analysis* 107, pp. 24–39.
- Gnedenko, B.V. and A.N. Kolmogorov (1954). *Limit distributions for sums of independent random variables*. Addison-Wesley. URL: <https://books.google.de/books?id=7qVyAQAACAAJ>.
- Mandelbrot, Benoit (1960). “The Pareto-Levy law and the distribution of income”. In: *International Economic Review* 1.2, pp. 79–106.
- Muirhead, Robb J (2009). *Aspects of multivariate statistical theory*. Vol. 197. John Wiley & Sons.
- Nolan, John (2003). *Stable distributions: models for heavy-tailed data*. Birkhauser New York.

Chapter 4

Variational Quantum Circuit Model for Knowledge Graph Embedding

Variational Quantum Circuit Model for Knowledge Graph Embedding

Yunpu Ma,* Volker Tresp, Liming Zhao, and Yuyi Wang

In this work, the first quantum Ansätze for the statistical relational learning on knowledge graphs using parametric quantum circuits are proposed. Two types of variational quantum circuits for knowledge graph embedding are introduced. Inspired by the classical representation learning, latent features for entities are first considered as coefficients of quantum states, while predicates are characterized by parametric gates acting on the quantum states. For the first model, the quantum advantages disappear when it comes to the optimization of this model. Therefore, a second quantum circuit model is introduced where embeddings of entities are generated from parameterized quantum gates acting on the pure quantum state. The benefit of the second method is that the quantum embeddings can be trained efficiently meanwhile preserving the quantum advantages. It is shown that the proposed methods can achieve comparable results to the state-of-the-art classical models, for example, RESCAL, DISTMULT. Furthermore, after optimizing the models, the complexity of inductive inference on the knowledge graphs might be reduced with respect to the number of entities.

1. Introduction

Over the last few years, some large-scale triple-oriented knowledge databases have been generated. These databases are principled approaches to knowledge representation and reasoning. They are widely used in large-scale artificial intelligence systems, such as question answering engines, human-computer interaction platforms, and decision-making support systems. One well-known example is the IBM's cognitive computing platform, the IBM Watson, where knowledge graphs are at the core of it. The

other example is the largest universally accessible knowledge graph (KG) maintained by Google.

Nowadays, knowledge graphs proliferate with increasing numbers of semantic triples and distinct entities. The reason is that knowledge graphs collect and merge information from various unstructured data, such as publications and internet. The increasing number of semantic triples and distinct entities leads to a slow training of knowledge graphs, as well as a sluggish response to the inductive inference tasks on knowledge graphs after training. Therefore, in order to accelerate the learning and inference on knowledge graphs, we propose statistical relational learning using quantum Ansätze.

In this work, we propose the first quantum Ansätze for modeling and learning large-scale relational databases using parametric quantum circuits. We simulate our quantum learning algorithms on graphics processing units (GPUs) and demonstrate the model performance on multiple state-of-the-art relational databases. We will also discuss how these quantum Ansätze could speed up the inference.

2. Representation Learnings on Knowledge Graphs

Various statistical relational models for large-scale KGs have been proposed in the literature, such as the bilinear model (RESCAL^[1]), the bilinear diagonal model (DISTMULT^[2]), the complex embedding model (COMPLEX^[3]). In this section, we first introduce knowledge graphs, and provide a succinct introduction to representation learning in KGs. We adapt the notation of ref. [4] for convenience.

2.1. Knowledge Graphs


Knowledge graphs are triple-oriented knowledge representations. The core components of KGs are semantic triples (*subject, predicate, object*) where subject and object are entities represented as nodes in the graph and where predicate indicates the labeled link from the subject to the object. One example of a semantic triple in Figure 1 could be (*Angela_Merkel, Chancellor_of, Germany*). Observed semantic triples (marked as solid lines in Figure 1) are elements of the training dataset, while unobserved triples (marked as dashed lines) will be inferred during the test.

Y. Ma, Prof. V. Tresp
Ludwig Maximilian University of Munich
Geschwister-Scholl-Platz 1, 80539 Munich, Germany
E-mail: yunpu.ma@siemens.com

Y. Ma, Prof. V. Tresp
Siemens AG
Otto-Hahn-Ring 6, 81739 Munich, Germany

Dr. L. Zhao
Singapore University of Technology and Design
Singapore

Dr. Y. Wang
ETH Zurich
Gloriastrasse 35, 8092 Zurich, Switzerland

 The ORCID identification number(s) for the author(s) of this article can be found under <https://doi.org/10.1002/qute.201800078>

DOI: 10.1002/qute.201800078

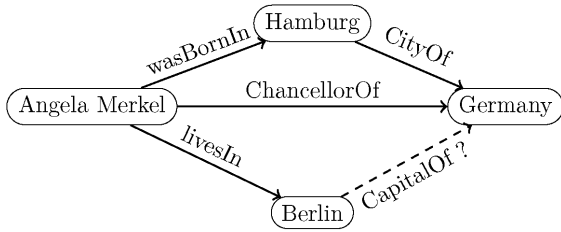


Figure 1. A knowledge graph fragment: observed semantic triples are marked with solid arrows, while unobserved semantic triples are marked with dashed arrows.

2.2. Representation Learning

Let \mathcal{E} denote the set of entities, and \mathcal{P} the set of predicates. Let N_e be the number of entities in \mathcal{E} , and N_p the number of predicates in \mathcal{P} . Given a predicate $p \in \mathcal{P}$, the indicator function $\phi_p : \mathcal{E} \times \mathcal{E} \rightarrow \{1, 0\}$ indicates whether a triple (\cdot, p, \cdot) is true or false. Furthermore, \mathcal{R}_p indicates the set of all subject–object pairs, such that $\phi_p = 1$. The entire KG can be written as $\chi = \{(i, j, k)\}$, with $i = 1, \dots, N_e$, $j = 1, \dots, N_p$, and $k = 1, \dots, N_e$. A knowledge graph can be equivalently treated as a $N_e \times N_p \times N_e$ -dimensional three-order tensor, and an entry indicates whether a semantic triple is true, false or unobserved.

We assume that each entity and predicate has a unique latent representation. Let \mathbf{a}_{e_i} , $i = 1, \dots, N_e$, be the representations of entities, and \mathbf{a}_{p_i} , $i = 1, \dots, N_p$, be the representations of predicates. Note that \mathbf{a}_{e_i} and \mathbf{a}_{p_i} could be real- or complex-valued vectors/matrices. Moreover, when we consider a concrete example, say (s, p, o) , we use \mathbf{a}_s , \mathbf{a}_p , and \mathbf{a}_o to represent the latent representation of the subject s , the predicate p , and the object o , respectively.

A probabilistic model for the knowledge graph χ is defined as $\Pr(\phi_p(s, o) = 1 | \mathcal{A}) = \sigma(\eta_{spo})$ for all (s, p, o) -triples in χ , where $\mathcal{A} = \{\mathbf{a}_{e_i}\}_{i=1}^{N_e} \cup \{\mathbf{a}_{p_i}\}_{i=1}^{N_p}$ denotes the collection of all embeddings; $\sigma(\cdot)$ denotes the sigmoid function; η_{spo} is the value function of latent representations \mathbf{a}_s , \mathbf{a}_p , and \mathbf{a}_o . Given a labeled dataset containing both false and true triples $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^m$, with $x_i \in \chi$, and $y_i \in \{-1, 1\}$, latent representations can be learned from a loss function. Commonly, one minimizes the regularized logistic loss function

$$\min_{\mathcal{A}} \sum_{i=1}^m \log(1 + \exp(-y_i \eta_{x_i})) + \lambda \|\mathcal{A}\|_2^2 \quad (1)$$

where m is the number of training samples, η_{x_i} is the value function for the semantic triple x_i , and λ is the regularization hyperparameter. Another commonly used loss function is the regularized mean squared error (MSE) loss

$$\frac{1}{m} \sum_{i=1}^m (y_i - \eta_{x_i})^2 + \lambda \|\mathcal{A}\|_2^2 \quad (2)$$

Note that the value function η_{spo} can be defined differently in different models. For instance, for the RESCAL^[1] model, entities are represented as unique R -dimensional real-valued vectors, $\mathbf{a}_{e_i} \in \mathbb{R}^R$, with $i = 1, \dots, N_e$, and predicates are represented as

$R \times R$ matrices, $\mathbf{a}_{p_i} \in \mathbb{R}^{R \times R}$, with $i = 1, \dots, N_p$. Moreover, the value function is defined as

$$\eta_{spo} = \mathbf{a}_s^T \mathbf{a}_p \mathbf{a}_o \quad (3)$$

For DISTMULT^[2] $\mathbf{a}_{e_i}, \mathbf{a}_{p_j} \in \mathbb{R}^R$, with $i = 1, \dots, N_e$, $j = 1, \dots, N_p$. The value function is defined as

$$\eta_{spo} = \langle \mathbf{a}_s, \mathbf{a}_p, \mathbf{a}_o \rangle \quad (4)$$

where $\langle \cdot, \cdot, \cdot \rangle$ denotes the tri-linear dot product.

For COMPLEX^[3] entities and predicates are complex-valued vectors $\mathbf{a}_{e_i}, \mathbf{a}_{p_j} \in \mathbb{C}^R$, with $i = 1, \dots, N_e$, $j = 1, \dots, N_p$. The value function for the COMPLEX model reads

$$\eta_{spo} = \Re(\langle \mathbf{a}_s, \mathbf{a}_p, \bar{\mathbf{a}}_o \rangle) \quad (5)$$

where the bar denotes complex conjugate, and \Re denotes the real part of a complex number.

For the TUCKER^[5] tensor decomposition model, entities and predicates are real-valued vectors, $\mathbf{a}_{e_i} \in \mathbb{R}^R$, with $i = 1, \dots, N_e$, and $\mathbf{a}_{p_j} \in \mathbb{R}^R$, with $j = 1, \dots, N_p$. Additionally, a global core tensor $\mathbf{W} \in \mathbb{R}^{R \times R \times R}$ is introduced. The value function for the TUCKER model reads

$$\eta_{spo} = \mathbf{W} \times_1 \mathbf{a}_s \times_2 \mathbf{a}_p \times_3 \mathbf{a}_o \quad (6)$$

Tensor models and compositional models are principled approaches for modeling large-scale relational data. The global relational patterns are encoded in the latent features of entities and predicates after optimizing the models. Thus, it is beneficial to analyze how the dimensionality of latent features influences the expressiveness and the generalization ability of the models. These questions have been studied in ref. [6]. In order to interpret the results in ref. [6], we first introduce the following notations.

Definition 1. Let $\mathbf{X} \in \mathbb{R}^{\prod_{i=1}^m n_i}$ be an m -order tensor with dimensions $\mathbf{n} = (n_1, \dots, n_m)$. Suppose that it can be written as a (Tucker) tensor product $\mathbf{X} = \mathbf{W} \times_1 U^{(1)} \times_2 \dots \times_m U^{(m)}$ with n -rank $\mathbf{R} = (R_1, \dots, R_m)$, where $\mathbf{W} \in \mathbb{R}^{\prod_{i=1}^m R_i}$ is the core tensor, and $U^{(i)} \in \mathbb{R}^{n_i \times R_i}$ are the latent factor matrices. Each entry of the tensor \mathbf{X} can be written as a polynomial

$$x_{i_1, \dots, i_m} = \sum_{j_1=1}^{R_1} \dots \sum_{j_m=1}^{R_m} w_{j_1, \dots, j_m} \prod_{k=1}^m u_{i_k, j_k}^{(k)}$$

The set of different sign patterns which can be expressed by the tensor \mathbf{X} is defined as

$$\mathcal{S}_{\mathbf{n}, \mathbf{R}} := \{\text{sgn}(\mathbf{X}) \in \{-1, 0, +1\}^{\prod_{i=1}^m n_i} | n\text{-rank}(\mathbf{X}) \leq \mathbf{R}\} \quad (7)$$

Note the cardinality $|\mathcal{S}_{\mathbf{n}, \mathbf{R}}|$ indicates how expressive and flexible the Tucker tensor decomposition could be. For the KGs modeling with tensor decomposition, we focus on the case of three-order tensors. The upper bound of $|\mathcal{S}_{\mathbf{n}, \mathbf{R}}|$ is given in the following Lemma.

Lemma 1 (Upper Bound for Sign Patterns^[6]). Consider a three-order tensor $\mathbf{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ which can be written as a tensor product $\mathbf{X} = \mathbf{W} \times_1 U^{(1)} \times_2 U^{(2)} \times_3 U^{(3)}$ with rank $\mathbf{R} = (R_1, R_2, R_3)$. The

number of different sign patterns of the tensor \mathbf{X} is upper bounded by the following number

$$|\mathcal{S}_{n,R}| \leq \left(\frac{16e n_1 n_2 n_3}{\text{var}(\mathbf{X})} \right)^{\text{var}(\mathbf{X})} \quad (8)$$

where $\text{var}(\mathbf{X})$ is defined as $\text{var}(\mathbf{X}) := \prod_{i=1}^3 R_i + \sum_{i=1}^3 n_i R_i$.

Given observed entries of a KG, the above Lemma indicates that the ranks should be large enough to fit the observed entries via the tensor decomposition. Therefore, in order to model an ever-increasing KG with increasingly complex relational structures, the dimension of latent features also needs to grow with the KG. As a reminder, the complexity of value functions grows at least linearly with the dimension of the latent features for entities. For example, the computational complexity of the value function for the **DISTMULT** model is $\mathcal{O}(R)$ (see Equation (4)), while for the **TUCKER** model it becomes $\mathcal{O}(R^3)$ (see Equation (6)). One goal of this work is to learn a probabilistic model for relational databases by making a quantum Ansatz for the value function. We will discuss how the evaluation of value functions can be accelerated via low-depth quantum circuits.

3. Quantum Circuit Models

In this section, we focus on variational unitary circuits. Algorithms of quantum classifiers using variational unitary circuits with parameterized and trainable gates have been proposed in ref. [7]. A quantum circuit U composed of L unitary operations can be decomposed into a product of unitary matrices

$$U = U_L \cdots U_1 \cdots U_1$$

where each U_i indicates either a unitary operation on one qubit or a controlled gate acting on two qubits. Since a single qubit gate is a 2×2 unitary matrix in $SU(2)$, we apply the following parameterization

$$G(\alpha, \beta, \gamma) = \begin{pmatrix} e^{i\beta} \cos \alpha & e^{i\gamma} \sin \alpha \\ -e^{-i\gamma} \sin \alpha & e^{-i\beta} \cos \alpha \end{pmatrix} \quad (9)$$

where $\{\alpha, \beta, \gamma\}$ are the tunable parameters of the single qubit gate. Note that a global phase factor is neglected.

In the following, we introduce the parameterization of controlled gates. The controlled gate $C_i(G_j)$ which acts on the j -th qubit conditioned on the state of the i -th qubit can be defined as

$$C_i(G_j) |x\rangle_i \otimes |y\rangle_j = |x\rangle_i \otimes G_j^x |y\rangle_j$$

where $|x\rangle_i, |y\rangle_j$ denotes the state of the i -th and the j -th qubit, respectively.

Using the parametric gates G and $C(G)$, we are capable to describe the quantum circuit model U_θ with parameterization θ in more details. Let us consider a quantum state with n entangled qubits. Suppose that the l -th unitary operation U_l is a single qubit gate acting on the k -th qubit, then it can be written as

$$U_l = \mathbb{1}_1 \otimes \cdots \otimes G_k \otimes \cdots \otimes \mathbb{1}_n$$

If the l -th unitary operation acts on the j -th qubit and conditioned on the state of the i -th qubit, U_l will have the following matrix representation

$$U_l = \mathbb{1}_1 \otimes \cdots \otimes \underbrace{\mathbb{P}_0}_{i\text{-th}} \otimes \cdots \otimes \underbrace{\mathbb{1}_j}_{j\text{-th}} \otimes \cdots \otimes \mathbb{1}_n \\ + \mathbb{1}_1 \otimes \cdots \otimes \underbrace{\mathbb{P}_1}_{i\text{-th}} \otimes \cdots \otimes \underbrace{G_j}_{j\text{-th}} \otimes \cdots \otimes \mathbb{1}_n,$$

$$\text{where } \mathbb{P}_0 = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \text{ and } \mathbb{P}_1 = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}.$$

4. Circuit Models for Knowledge Graphs

In this section, we introduce two quantum Ansätze for the value function and compare their computational complexities.

4.1. Quantum Circuit Embedding

We first introduce the *Quantum Circuit Embedding* (QCE) model, which can be considered as a generalization of the classical RESCAL model to the quantum regime. Similar to the RESCAL model, in QCE entities are represented by R -dimensional latent features. Without loss of generality, we assume that $R = 2^r$. In this way, an R -dimensional latent vector corresponds to a state of an r -qubit system.

One significant barrier to quantum learning algorithms is an efficient preparation of quantum states from classical data. In this work, we only consider real-valued representations for entities stored in a classical data structure \mathcal{T} . Then, a technique developed in ref. [8] can be utilized now, which can efficiently prepare the quantum states corresponding to latent features by accessing the classical data structure \mathcal{T} . In this way, the complexity of quantum state preparation can be reduced to the logarithm of R . Details related to the classical data structure \mathcal{T} and the preparation of quantum states via \mathcal{T} are relegated to the appendix. In summary, in the QCE model, entities are defined as $\mathbf{a}_{e_i} \in \mathbb{R}^R$, with normalized l_2 -norm $\|\mathbf{a}_{e_i}\|_2 = 1$, for $i = 1, \dots, N_e$.

Furthermore, in QCE each predicate p is associated with a specific quantum circuit composed of sequential implementations of variational gates. Therefore, each predicate has an $R \times R$ unitary matrix representation $U_p(\theta_p)$, where θ_p are the predicate-specific trainable parameters stemming from the variational quantum gates. Moreover, we fix the circuit architecture of implementing predicates such that each predicate is uniquely determined by the circuit parameterizations θ_p .

Given a semantic triple (s, p, o) , how the value function η_{spo} is defined in the quantum model? As a reminder, in The RESCAL model, the value function η_{spo} can be seen as the dot product of two vectors \mathbf{a}_{sp} and \mathbf{a}_o , where $\mathbf{a}_{sp} := \mathbf{a}_s^\top \mathbf{a}_p$. The loss function encourages the two vectors \mathbf{a}_{sp} and \mathbf{a}_o to point in the same direction if the given semantic triple is genuine, otherwise in opposite directions.

Inspired by the classical model **COMPLEX**, we define the quantity $\eta_{spo}^{\text{QCE}} := \Re \langle o | U_p(\theta_p) | s \rangle$. This quantity is the real part of the

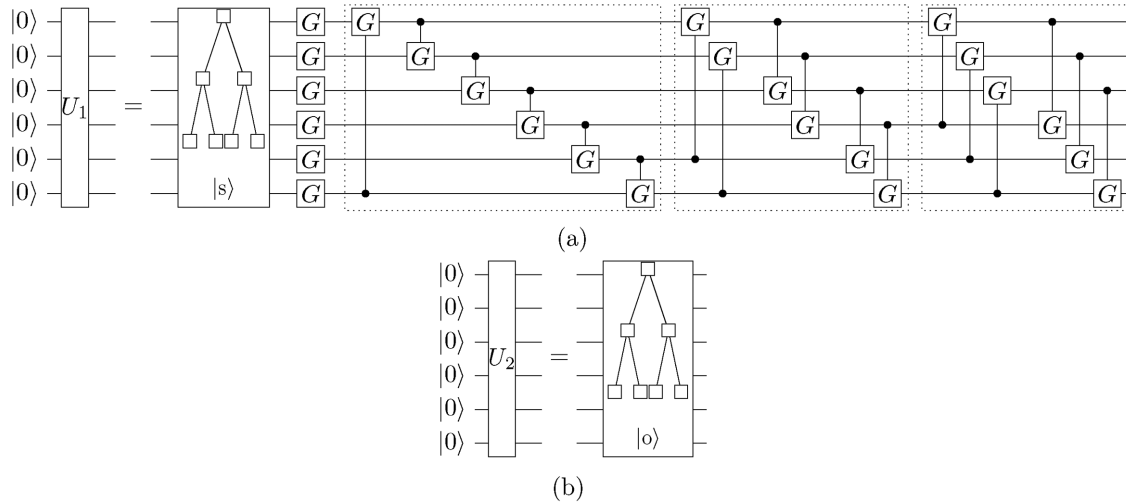


Figure 2. Building blocks of the QCE model. a) The U_1 module in the QCE model is displayed. U_1 encodes the latent feature of the subject s as the state $|s\rangle$. The quantum circuit associated to the predicate p maps the ket state $|s\rangle$ to another ket state $U_p(\theta_p) |s\rangle$. For all the following experiments, we set the dimension of entity latent features as $R = 64$, which corresponds to a six-qubit system. In addition, the circuit architecture for all predicates is fixed, and it can be decomposed in four blocks: single qubit gates, two-qubit controlled gates with control range 1, 2, and 3 (dashed blocks). b) The U_2 module in the QCE model, which prepares the quantum state $|o\rangle$ is displayed. In both (a) and (b), the tree structure represents the quantum access to the classical data structure \mathcal{T} .

inner product of two quantum states $|o\rangle$ and $|sp\rangle := U_p(\theta_p) |s\rangle$ generated by separate unitary circuits. The model parameters can be optimized by maximizing the inner product given genuine triples and minimizing the inner product given false or unobserved semantic triples. A relation between η_{spo}^{QCE} and the label of the triple (s, p, o) will be specified later.

We explain the circuit architecture in more details. Latent features \mathbf{a}_s for the subject and \mathbf{a}_o for the object are first encoded in quantum states $|s\rangle$ and $|o\rangle$ through a quantum access to the memory structure \mathcal{T} . The dimension of features is set to $R = 64$ in the following experiments, which corresponds to a six-qubit system. Next, a unitary circuit $U_p(\theta_p)$ corresponding to the predicate p evolves $|s\rangle$ to the state $U_p(\theta_p) |s\rangle$. Note that both the latent features of entities and the parametric circuits of predicates need to be optimized during the training.

We develop the circuit for predicates out of four building blocks, and each block consists of variational gates or controlled gates operating on each of the six qubits. To be more specific, the first block consists of single qubit rotations, and the rest of the blocks consist of two-qubit controlled gates with control range 1, 2, 3, respectively. So, the unitary circuit associated with the predicates can be written as $U_{p_i}(\theta_{p_i}) = U_4 U_3 U_2 U_1$, with $i = 1, \dots, N_p$, where

$$\begin{aligned} U_1 &= G_6 G_5 G_4 G_3 G_2 G_1 \\ U_2 &= C_6(G_1) C_1(G_2) C_2(G_3) C_3(G_4) C_4(G_5) C_5(G_6) \\ U_3 &= C_5(G_1) C_6(G_2) C_1(G_3) C_2(G_4) C_3(G_5) C_4(G_6) \\ U_4 &= C_4(G_1) C_5(G_2) C_6(G_3) C_1(G_4) C_2(G_5) C_3(G_6) \end{aligned} \quad (10)$$

Note that the index for the predicate was neglected since we assume that the circuit architecture is fixed for all the predicates. **Figure 2** illustrates the circuits for preparing the states $|o\rangle$ and

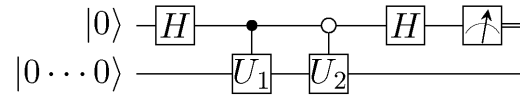


Figure 3. Quantum circuit for estimating the value $\Re \langle o | U_p(\theta_p) | s \rangle$. The detailed architectures of unitary evolutions U_1 and U_2 can be found in Figure 2 for the QCE model and Figure 4 for the rQCE model.

$|sp\rangle$. In the following, we show that the value η_{spo}^{QCE} can be measured physically. We adopt a similar idea to SWAP test for discriminating two quantum states. The SWAP test was initially proposed for quantum fingerprinting,^[9] and it was further developed within refs. [10,11] for discriminating quantum evolution operators.

The basic idea is illustrated in **Figure 3**. This architecture is inspired by ref. [11] and Observation 3 in ref. [7]. Consider two unitary operations U_1 and U_2 which operate on a pure state $|0\rangle := |0 \dots 0\rangle$ conditioned on the state of the ancilla qubit. Particularly, the quantum state becomes $U_1 |0\rangle$ if the ancilla qubit is $|1\rangle_A$ and $U_2 |0\rangle$ if it is in the state $|0\rangle_A$. Before measuring the ancilla qubit, the underlying quantum state of the entire system reads

$$\frac{1}{\sqrt{2}} (|0\rangle_A U_2 |0\rangle + |1\rangle_A U_1 |0\rangle)$$

The second Hadamard gate acting on the ancilla qubit brings the state to

$$\frac{1}{2} [|0\rangle_A (U_2 |0\rangle + U_1 |0\rangle) + |1\rangle_A (U_2 |0\rangle - U_1 |0\rangle)]$$

For the QCE model, the unitary modules U_1 and U_2 are illustrated in Figure 2. Considering a concrete semantic triplet (s, p, o) , with an access to the quantum gates parameters we can prepare the following quantum state

according to the second Hadamard formula above:

$$\frac{1}{2} [|0\rangle_A (|o\rangle + |sp\rangle) + |1\rangle_A (|o\rangle - |sp\rangle)]$$

Therefore, the probability of sampling the ancilla qubit in the state $|0\rangle_A$ is

$$\Pr(|0\rangle_A) = \frac{1}{2} + \frac{1}{2} \Re \langle o | sp \rangle = \frac{1}{2} + \frac{1}{2} \eta_{spo} \quad (11)$$

while the probability of measuring it in the state $|1\rangle_A$ is

$$\Pr(|1\rangle_A) = \frac{1}{2} - \frac{1}{2} \Re \langle o | sp \rangle = \frac{1}{2} - \frac{1}{2} \eta_{spo} \quad (12)$$

In the upper equation, we temporarily neglect the superscript of the value function. As we can see, the value η_{spo} is related to the statistics of sampled quantum states of the ancillary qubit via

$$\eta_{spo} = 2 \Pr(|0\rangle_A) - 1 = 1 - 2 \Pr(|1\rangle_A) \quad (13)$$

Similar to the classical models, this quantity defines the loss function jointly with the labels of the triplets.

4.2. Loss Function and Training

Details of the loss function and the optimization method are provided in this section. We focus on the QCE model. Given a training dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^m$ with $x_i \in \mathcal{X}$, the loss function of the quantum circuit model is defined as the following mean error

$$\mathcal{L} = \frac{1}{m} \sum_{i=1}^m (y_i - \eta_{x_i}^{\text{QCE}})^{2\kappa} \quad (14)$$

where $y_i \in \{-1, 1\}$ are labels, and $\kappa \in \mathbb{Z}^+$ is a hyperparameter. The reason for this choice of the labels will be clarified later. One can also notice that for the quantum model, the loss function is not regularized by the norm of parameters. Because of the unitary constraint on the evolution of quantum circuits, hidden quantum states are automatically normalized. Therefore, the l_2 norm cannot either effect the norms of embedding vectors or improve the generalization ability of the quantum circuit model.

With the loss function, the model is optimized by updating the parameters via gradient descent. Parameters of the variational gates can be efficiently estimated using a hybrid gradient descent scheme introduced in ref. [7]. The partial derivative of Equation (14) with respect to the gate parameters reads

$$\frac{\partial \mathcal{L}}{\partial \theta} = \frac{2\kappa}{m} \sum_{i=1}^m (\eta_{x_i}^{\text{QCE}} - y_i)^{2\kappa-1} \frac{\partial}{\partial \theta} \eta_{x_i}^{\text{QCE}} \quad (15)$$

where $\theta \in \{\alpha_{p_i}, \beta_{p_i}, \gamma_{p_i}\}$, with $i = 1, \dots, N_p$.

The techniques developed within refs. [7,12] allow the above partial derivative to be estimated from the states' statistics of the ancilla qubit since the partial derivative can be written as a linear combination of gates with shifted parameters. To be specific, we

have the following derivatives for a single qubit gate

$$\frac{\partial}{\partial \alpha} G(\alpha, \beta, \gamma) = G\left(\alpha + \frac{\pi}{2}, \beta, \gamma\right)$$

$$\frac{\partial}{\partial \beta} G(\alpha, \beta, \gamma) = \frac{1}{2} G\left(\alpha, \beta + \frac{\pi}{2}, 0\right) + \frac{1}{2} G\left(\alpha, \beta + \frac{\pi}{2}, \pi\right)$$

$$\frac{\partial}{\partial \gamma} G(\alpha, \beta, \gamma) = \frac{1}{2} G\left(\alpha, 0, \gamma + \frac{\pi}{2}\right) + \frac{1}{2} G\left(\alpha, \pi, \gamma + \frac{\pi}{2}\right)$$

Moreover, partial derivatives of two-qubit gates can be written as a combination of control gates with shifted gates' parameters. More details of the hybrid gradient descent approach can be found in Section 4 of ref. [7].

However, this technique cannot be applied to the estimation of the gradients with respect to the latent features of entities. Another problem is that even if we could efficiently estimate the gradients with respect to the latent features, the entire classical data structure \mathcal{T} needs to be updated after each step of optimization due to the normalization constraints. It leads to a computational overhead of $\mathcal{O}(R^2)$ for just one update of \mathbf{a}_{e_i} , with $i = 1, \dots, N_e$.

4.3. Fully Parameterized Quantum Circuit Embedding

To overcome the disadvantages of the QCE, at this place, we introduce another *fully parameterized Quantum Circuit Embedding* (FQCE) model. The idea behind FQCE is reasonably simple. Instead of storing and reading entity features as normalized R -dimensional vectors, they are obtained by applying parameterized quantum circuit to initial quantum states which can be easily prepared. In this way, each entity is uniquely identified by the circuit architecture and the gates parameters similar to the circuits definition of predicates in the QCE model.

Compared to the QCE model, the advantages of this approach are twofolds. First, latent features do not need to be loaded from the classical data structure \mathcal{T} and encoded as the coefficients of quantum states. Alternatively, they are generated from the quantum evolution of initial quantum states. Second, FQCE can be optimized efficiently since the only trainable parameters are in the variational quantum gates. Therefore, techniques explained in the last subsection can be applied to accelerate the optimization.

The circuit architecture for generating quantum representations of entities is given in **Figure 4** and overall we fix the circuit architecture for all entities. The six-qubit quantum system is initialized as a pure quantum state $|0\rangle$. Hadamard gates act on each qubit to create a superposition $H_{6,\dots,1}|0\rangle := H_6 H_5 \dots H_1 |0\rangle$. Subsequently, an entity-specific unitary circuit develops the quantum representation from the superposition,

$$|e_i\rangle = U_{e_i} H_{6,\dots,1} |0\rangle, \quad \text{with } i = 1, \dots, N_e \quad (16)$$

where U_{e_i} have the same circuit architecture design as U_{p_i} in Equation (10).

To harvest the quantum advantages, the circuit depth should be low and in the order of $\log(R)$. In this way, we only need to replicate the experiments $\mathcal{O}(\log^2 R / \epsilon^2)$ times to update the model parameters given one training example, where ϵ is the accuracy

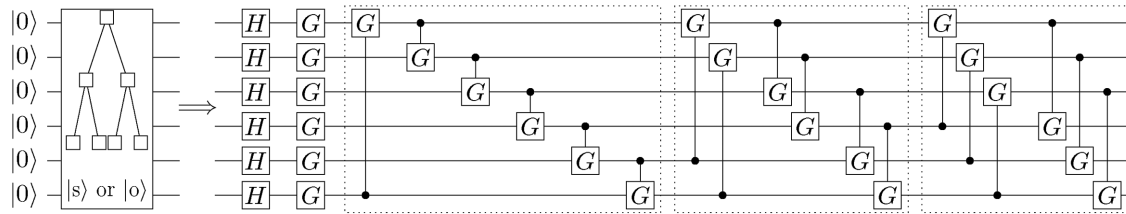


Figure 4. In the FQCE model, the classical data structure \mathcal{T} is replaced by variational unitary circuits. Therefore, the quantum states $|s\rangle$ or $|o\rangle$ can be prepared by applying unitary circuits to the initial quantum states $|00\cdots 0\rangle$, instead of loading data from the classical data structure \mathcal{T} . Note that the circuit architecture is fixed for all entities (subjects and objects). The unitary circuit contains five blocks. The first block consists of Hadamard gates which can develop superposition from the initial quantum state $|0\cdots 0\rangle$. The rest of the blocks consist of single qubit gates and two-qubit controlled gates with control range 1, 2, and 3 (dashed blocks).

required. Moreover, the overall model architecture for estimating the value function η_{spo}^{FQCE} using ancilla qubit remains the same as in Figure 3.

Before simulating the proposed quantum Ansätze, we compare computational complexities of them. We first consider the time complexity of evaluating the value function. In the QCE model, loading the entity features from the classical data structure \mathcal{T} costs time $\mathcal{O}(\log R)$. Since we use shallow circuits with depth $\mathcal{O}(\log R)$ to specify the predicates, the unitary evolution of quantum states for entities requires $\mathcal{O}(\log^2 R)$ unitary operations. The value function is estimated from the Bernoulli distribution of the ancilla qubit. Therefore, one has to perform $\mathcal{O}(\frac{1}{\epsilon^2})$ repetitions of the experiment in Figure 3 to resolve the statistics of the ancilla qubit up to a predefined error ϵ . To summarize, the entire procedure of evaluating η_{spo}^{QCE} can be completed in runtime $\mathcal{O}(\text{poly}(\log R, \frac{1}{\epsilon}))$. Similarly, the evaluation of η_{spo}^{FQCE} requires a runtime $\mathcal{O}(\text{poly}(\log R, \frac{1}{\epsilon}))$.

A notable complexity difference between two quantum circuit models appears when it comes to the training. Let us first consider the FQCE model. Given one training sample (s, p, o) , it requires $\mathcal{O}(\log^2 R/\epsilon^2)$ repetitions of the experiments to estimate the gradients and update the parameters in U_s , U_p , and U_o . Let D indicate the total number of semantic triples in the training dataset, then the runtime of one epoch is $\mathcal{O}(D \text{ poly}(\log R, \frac{1}{\epsilon}))$. However, for QCE, the runtime of one training epoch becomes $\mathcal{O}(D \text{ poly}(R, \log R, \frac{1}{\epsilon}))$ since after each step of optimization, re-normalizing the entity latent features and updating the classical memory structure \mathcal{T} require additional $\mathcal{O}(R)$ operations. As one can see, the quantum advantages disappear when we optimize the QCE model.

5. Experiments

5.1. Datasets and Evaluation

To evaluate proposed quantum models for knowledge graph embedding, we use four link prediction datasets of different sizes: KINSHIP,^[13] FB15K-237,^[14] WN18RR,^[15] and GDELT.^[16]

KINSHIP contains relations between family members. An example of the triple is *(Max, husband_of, Mary)*.

FB15K-237 is a subset of FREEBASE with only 237 predicates. Most of the semantic triples in the FB15K-237 are related to the facts of cities, movies, sports, and musics, for example, *(California, located_in, U.S.)*.

Table 1. Statistics of different knowledge graphs.

	# \mathcal{D}	N_e	N_p	N_a
KINSHIP	10 790	104	26	≈ 104
WN18RR	79 043	39 462	18	≈ 2
FB15K-237	310 079	14 505	237	≈ 21
GDELT	497 603	6785	230	≈ 73

GDELT The Global Database of Events, Language and Tone (GDELT) monitors events between different countries and organizations. An example could be *(ESA, collaborates with, NASA)*.

WN18RR This hierarchical knowledge base is a subset of WORDNET which consists of hyponym and hypernym relations between words, for example, *(pigeon, hyponym_of, bird)*.

The exact statistics of datasets are listed in Table 1, including the total number of triplets in the dataset $\#D$, the number of entities N_e , the number of predicates N_p , and the average number of labeled links connecting to a node N_a .

Since the above-mentioned datasets only consist of positive (genuine) semantic triples, we generate negative (false) instances according to the method of corrupting semantic triples proposed in ref. [17]. Given a genuine semantic triple (s, p, o) , negative triples are drawn by corrupting the object o to a different entity o' , and similarly corrupting subject s to s' . This corruption method makes a local-closed world assumption, meaning that the knowledge graph is assumed to be only locally connected. Therefore, corrupted and unobserved semantic triples are treated as negative examples during the training.

The model performance is evaluated using the following metrics on the test dataset. Let us consider a semantic triple (s, p, o) in the test dataset. To evaluate the retrieval of the object o , given the subject s and the predicate p , we first replace the object o with every object o' and compute the values of $\eta_{spo'}$. Following that, we sort these values in a decreasing order and locate the target object o . This position is referred to as the rank of the target object. We provide the filtered ranking scores as suggested in ref. [17] by removing all semantic triples (s, p, o') with $\gamma_{spo'} = 1$ and $o' \neq o$. Filtered ranking scores eliminate the ambiguity of retrieved information and provide a clearer performance evaluation of different models. In the same way, we also evaluated the retrieval of the subject s by ranking $\eta_{s'po}$ for all possible subjects s' .

To quantify the performance of the classical and quantum models on missing links predication, we report three metrics: filtered mean rank (MR), filtered Hits@3, and filtered Hits@10

Table 2. Filtered recall scores evaluated on four different datasets. Four metrics are compared: filtered Mean Rank (MR), filtered Hits@3 (@3), and filtered Hits@10 (@10).

Methods	KINSHIP			WN18RR			FB15k-237			GDEL T		
	MR	@3	@10	MR	@3	@10	MR	@3	@10	MR	@3	@10
RESCAL	3.2	88.8	95.5	12036	21.3	25.0	291.3	20.7	35.1	185.0	10.4	22.2
DISTMULT	4.5	61.0	87.7	10903	21.0	24.8	305.4	23.4	39.1	130.4	12.1	24.5
TUCKER	2.9	89.8	95.0	11997	19.1	23.9	276.1	20.9	35.7	144.0	14.5	27.3
COMPLEX	2.2	90.0	97.7	11895	24.6	26.1	242.7	25.2	39.7	137.6	12.9	26.4
<i>Best Known</i>	–	–	–	4187 ^[15]	44.0	52.0	244.0 ^[15]	35.6	50.1	102.0 ^[20]	31.5	47.1
QCE	3.6	73.8	93.8	3655	19.5	32.3	258.7	22.5	35.0	128.8	12.5	23.8
ƒQCE	3.6	73.1	94.0	2160	27.4	37.8	236.0	19.8	33.7	131.0	10.8	24.1

evaluated on the test dataset. Filtered mean rank is the average filtered ranking scores, and filtered Hits@n indicates the probability of finding the correct retrieval within the top-*n* filtered ranking.

The dimension of latent representations for all classical baselines is chosen as $R = 64$. For comparison, circuits algorithms for knowledge graph embedding are evaluated via six-qubit Ansätze. Overall, we fix the quantum circuit architecture depicted in Figure 2 for QCE and Figure 4 for ƒQCE model. Experiments show the recall scores on the test dataset are not sensitive to the order of implementing different blocks. Thus, for a simple implementation, we only consider four different blocks without repetitions. Exploration of various quantum circuit architectures to achieve better results could be an interesting research direction, and we leave it for future work.

During the training, the datasets are randomly split into training, validation, and test datasets. Early stopping on the validation set is used to avoid overfitting by monitoring the filtered Hits@3 entity recall scores every 20 epochs. Before training, all model parameters, including the entity embeddings and the gates parameters, are randomly initialized. In particular, we found that for the quantum Ansätze the model performance is relatively sensitive to the initialization of the gates parameters. After hyperparameter search, the gates parameters are initialized uniformly in the interval $[-\frac{\pi}{10}, \frac{\pi}{10}]$.

Here, we give more details on how quantum Ansätze are simulated. As discussed in Section 3, unitary evolution of a quantum state is equivalent to the unitary matrix-vector product. Therefore, we can simulate the quantum Ansätze on a single Tesla K80 GPU without exploiting real quantum devices.^[18] For the QCE model, each entity embedding is randomly initialized from a multivariate normal distribution and normalized after initialization. Embeddings for entities are stored as NumPy arrays instead of in the classical data structure \mathcal{T} . All the parameters, including entity embeddings and gate parameters, are updated via stochastic gradient descent. Moreover, after each step of the update, entity embeddings will be normalized again. Differently, for the ƒQCE model, each entity is initialized as $\frac{1}{8}|00\dots 0\rangle \equiv \frac{1}{8}(0, 0, \dots, 0)^T$ since all the trainable parameters are in the variational circuit. The codes are based on Tensorflow,^[19] and they will be available online.

Table 2 reports the performance of classical models and quantum Ansätze evaluated on different datasets. In addition, the

row *best known* in Table 2 shows the best results reported in the literatures.^[21] From the table, we can read that both quantum circuit models can achieve comparable results to the classical models using the dimension $R = 64$. In some cases, for example, the filtered Mean Rank recall scores on the WN18RR, FB15k-237, and GDEL T datasets, the quantum models can outperform all classical models.

Another interesting observation is that the Mean Rank score on the WN18RR dataset returned by the ƒQCE model is even better than the best-known models. We have to emphasize that among the four datasets, WN18RR contains the largest number of distinct entities (see Table 1). Therefore, ƒQCE is the desired quantum Ansatz of relational learning which shows both quantum advantages and superior performance on a vast database. However, one has to note that WN18RR possesses the smallest number of average links per node. Thus, questions are: Whether the quantum circuit models are only practical for modeling large and sparse datasets due to the intrinsic linearity of the circuit models; and whether applying nonlinearity activation functions on the circuit models^[22,23] can further improve the performance on other dense datasets? We leave these questions for future research.

5.2. Regularizations

As mentioned before, the quantum circuit models cannot be regularized using the l_2 -norm due to the unitarity constraints. Hence, what regularization methods can be applied to improve the generalization ability of the circuit models? We examine two techniques that are widely used in classical learning: dropout and Gaussian noise of model parameters. Generally speaking, dropout reduces the overfitting on the training dataset and noise encourages the model to land on flat minima of the loss surface. These two methods can be efficiently applied without destroying the unitarity restrictions.

We first apply the dropout technique. During the training, each quantum gate has a nonzero probability to be switched off. From the perspective of a classical neural network, this dropout is equivalent to randomly removing some weight matrices and replacing them with identity matrices. Similar regularization methods have been used to train very deep neural networks with vanishing gradients.^[24] However, all the gates will be

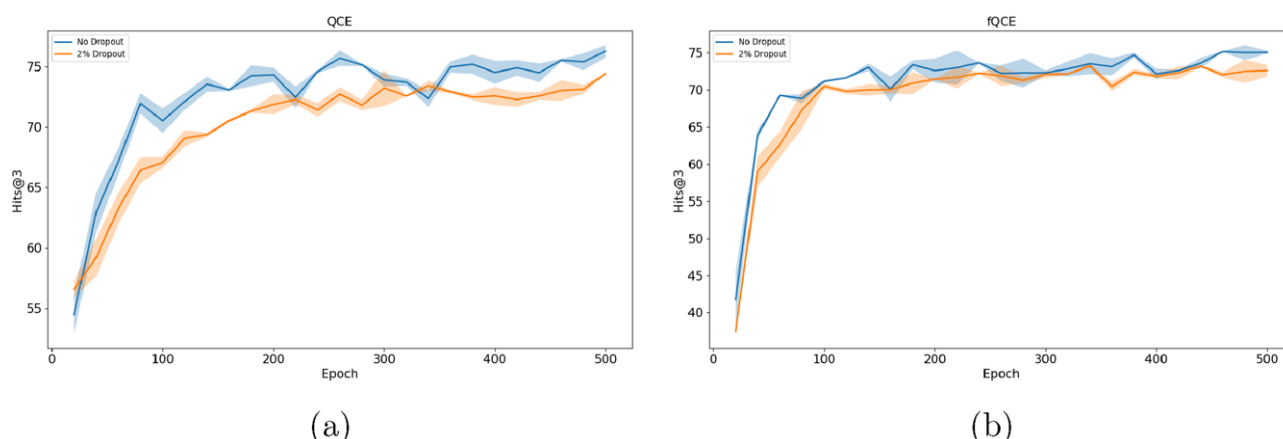


Figure 5. Comparison of the filtered Hits@3 recall scores on the KINSHIP dataset for a) QCE and b) FQCE. Blue line: without employing the dropout; orange line: 2% probability of dropping out a quantum gate randomly.

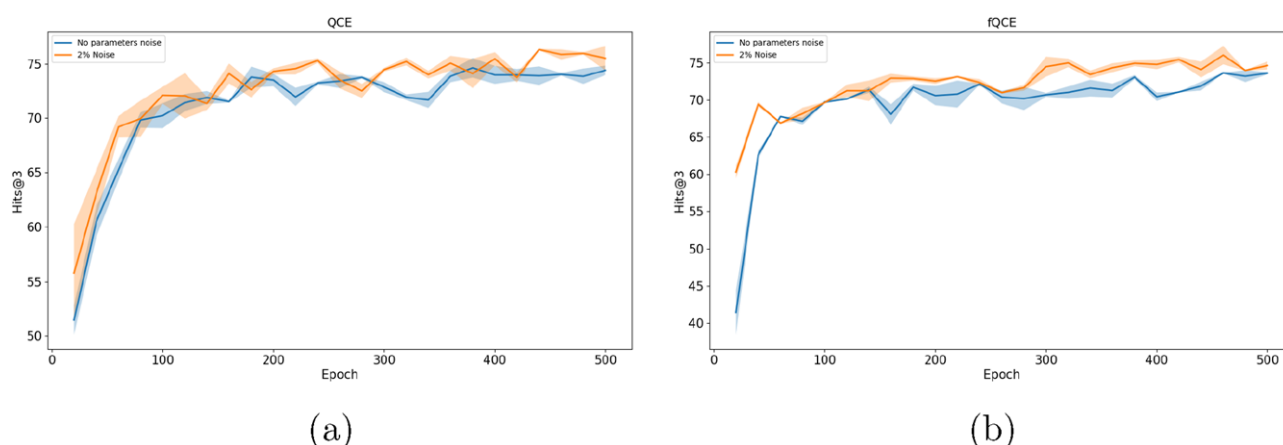


Figure 6. Filtered Hits@3 scores on the KINSHIP dataset for a) QCE and b) FQCE. Blue line: without introducing random noise; orange line: adding 2% noise to the model parameters both during the training and test.

implemented during the test phase. Because of the imperfections of quantum gates, quantum circuit models inherently possess this regularization property.

Simulations are performed for both circuit models using the KINSHIP dataset, and the dropout probability is chosen from $\{0.02, 0.05, 0.1, 0.2\}$. However, we could not observe any improvement in the performance, even using the smallest dropout probability. Recall scores for no dropout and 2% dropout probability are compared in **Figure 5**. Even though the dropout regularization cannot augment the performance of both models, we still learn that the FQCE model is more robust and resistant to imperfect quantum circuits, making it a potential candidate for the future test on real quantum devices.

Now we turn to study another regularization method which adds Gaussian noise to the model parameters. System noises are quite common in quantum computational devices, for example, they can stem from the disentanglement, flips of the qubits, or random phase rotations. However, in this work, we focus on noises stemming from inaccuracies. For example, the inevitably inaccurate loading of the classical data into quantum devices, the inaccurate parametric gates, or the statistical uncertainty about the state of ancilla qubit. To simulate quantum system imprec-

sion, we add Gaussian perturbations to the model parameters as follows

$$\theta' = \theta + \mu \mathcal{N}(0, |\theta|) \quad (17)$$

where θ could be a gate parameter or an element of an entity latent feature defined in the QCE model, and μ indicates the noise level. We further assume the amplitude of perturbation added to a model parameter is proportional to this parameter's absolute value.

To be more realistic, perturbations are introduced not only during the training but also in the *test phase*. **Figures 6** and **7** compare the recall scores, the filtered Mean Rank and filtered Hits@3, on the KINSHIP dataset. Performance improvement can be observed in both quantum models which indicates that system imprecision brings the models to flat minima of the loss functions. As first pointed out in ref. [25], the flatness of the minimum on the loss surface found by an optimization algorithm is a good indicator of the generalization ability. Improved performance by adding noise also suggests that the computational complexity can be reduced by controlling the accuracy ϵ .

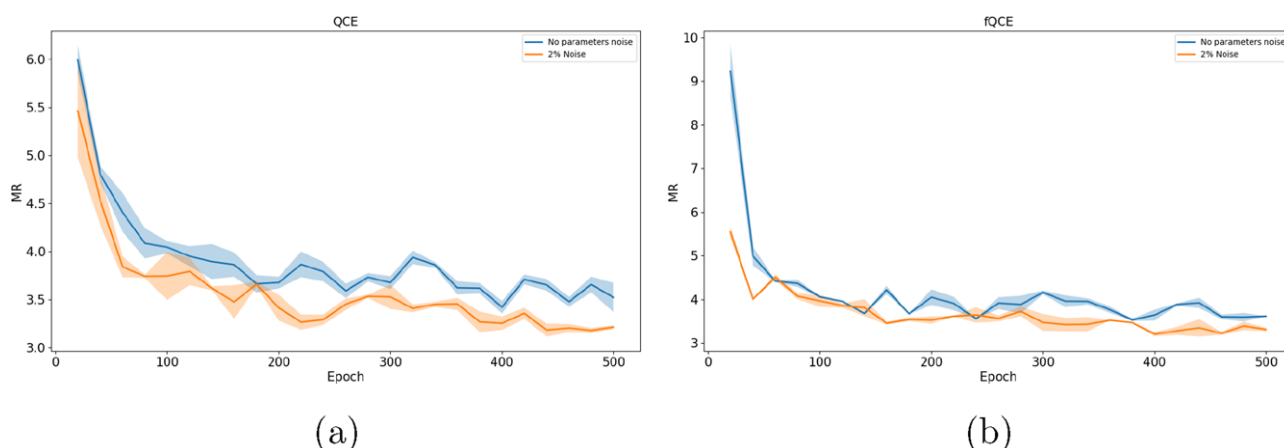


Figure 7. Filtered Mean Rank recall scores on the KINSHIP dataset for a) QCE and b) fQCE. Blue line: without introducing random noise; orange line: adding 2% noise to the model parameters both during the training and test.

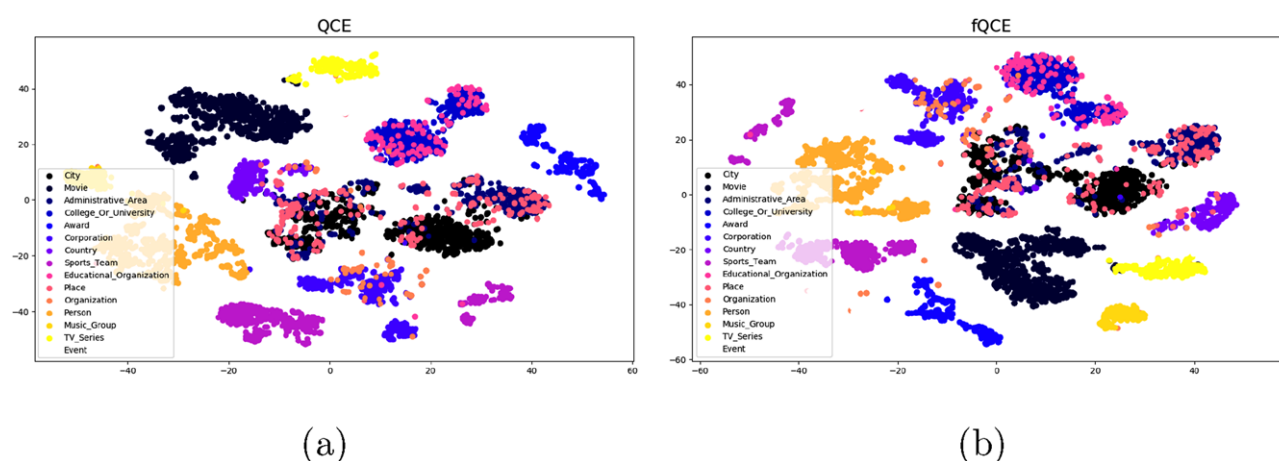


Figure 8. T-SNE visualizations of entity representations learned by a) QCE and b) fQCE.

5.3. T-SNE

We perform a qualitative analysis to visualize and understand the learned representations from the quantum Ansätze. Particularly, we focus on the latent features of entities. It has been reported that classical embeddings of entities show clustering effects. Entities with similar semantic meaning tend to group in the vector space. Here, we use t-SNE to analyze whether entity representations in the quantum models render this property. T-SNE^[26] is a powerful method for visualizing high-dimensional data on a 2D plane.

In order to visualize the semantic clustering effect, we focus on the FB15K-237 dataset, since it contains categorical information about the entities. We extract the top-15 most frequent categories, for example, *Movie*, *Administrative_Area*, *Organization*, and display them using different colors on the t-SNE plot. We still need to clarify how the quantum features are defined. In QCE, entity representations are normalized vectors $\mathbf{a}_{e_i} \in \mathbb{R}^R$, with $i = 1, \dots, N_e$. Besides, in the fQCE model, we define the hidden quantum states $|e_i\rangle = U_{e_i} H_{6,\dots,1} |0\rangle$, with $i = 1, \dots, N_e$ (see Equation (16)), as entity representations.

The t-SNE visualizations of learned quantum representations are displayed in **Figure 8**. One can clearly recognize the clustering effect based on the categories of entities. It is intriguing to point out that in **Figure 8**, the pink nodes representing the category *Educational_Organization* overlap with the blue nodes which represent the category *College_Or_University*.

Quantum circuit models reveal better semantic clustering effects of the learned latent features than classical models. **Figure 9** displays the t-SNE visualization of entity latent representations learned by DISTMULT. Particularly, one can notice that the learned latent features of the semantic categories *City*, *Administrative_Area*, and *Place* strongly overlap without revealing more detailed structures. The better semantic clustering effect might explain why fQCE performs consistently well when comparing with the Mean Rank metric.

6. Accelerated Inference

In previous sections, we have shown that the value functions can be evaluated with reduced complexity using the quantum

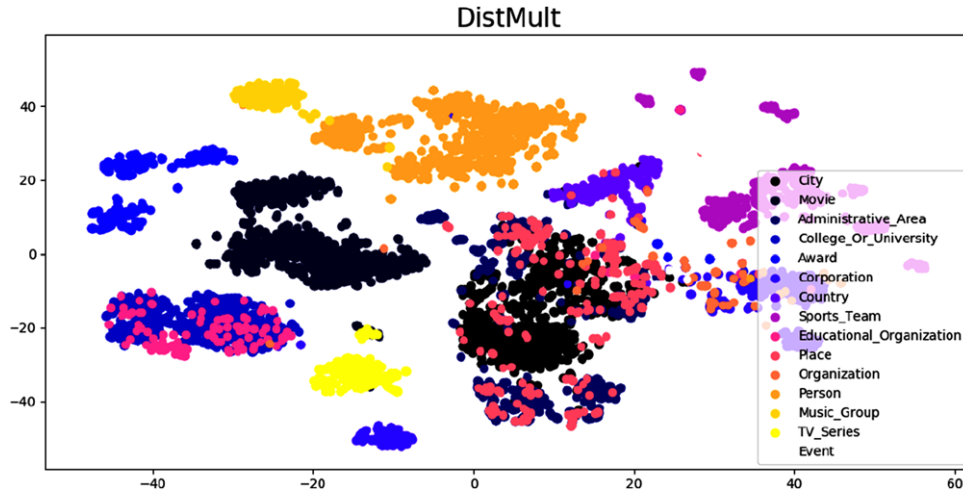


Figure 9. T-SNE visualization of entity representations learned by DISTMULT.

Ansätze. However, there is another quantum advantage we have not mentioned yet, namely the acceleration on the inference task. To be more specific, given an incomplete semantic triple (s, p, \cdot) , we attempt to find a quantum algorithm which can accelerate the search for the best (or the most possible) candidates for the unknown object.

What makes this task very challenging? As mentioned before, we are dealing with ever-increasing knowledge graphs with the consistently increasing number of distinct entities. Inference using classical models, for example, RESCAL and TUCKER requires many computation resources. The reason is we need to calculate all the value functions η_{spe_i} , with $i = 1, \dots, N_e$. Then, the entity e_i that corresponds to the maximum η_{spe_i} will be located, and the algorithm returns e_i as the best candidate for the unknown object. It could be extremely time and resource consuming since the same algorithm has to be repeated at least N_e times and each time of evaluation requires $\mathcal{O}(\text{poly } R)$ classical operations.

We are motivated to find a quantum algorithm showing quantum acceleration on the inference task. Here, we describe an *idealistic* and *heuristic* quantum algorithm for the inference. First, we prepare the following quantum state

$$\frac{1}{\sqrt{2N_e}} \sum_{i=1}^{N_e} (|0\rangle_A |i\rangle_I |0\rangle_L + |1\rangle_A |i\rangle_I |0\rangle_L) \quad (18)$$

The first qubit with the subscript A is an ancilla qubit. The second index register with the subscript I consists of $n_e := \lceil \log_2 N_e \rceil$ qubits, and the state $|i\rangle_I$ is the binary representation of the index i of the entity e_i . Furthermore, the third register with $r = \log_2 R$ qubits is prepared in the pure state $|0\rangle_R$ which will be used to generate the quantum representations of the entities.

Afterward, we use unitary circuit evolutions to prepare the states $|sp\rangle$ and $|e_i\rangle$. To be more specific, the U_1 circuit brings $|0\rangle_L$ to the state $|sp\rangle$ conditioned on the ancilla qubit being $|1\rangle_A$. Moreover, an entity-dependent circuit $U_2(e_i)$ brings $|0\rangle_L$ into the quantum state $|e_i\rangle$ conditioned on the ancilla being $|0\rangle_A$ and the index register being $|i\rangle_I$. Recall that the circuits U_1 and U_2 are defined in Figures 2 and 4.

To summarize, the unitary circuits will generate the following quantum state

$$\begin{aligned} & \frac{1}{\sqrt{2N_e}} \sum_{i=1}^{N_e} (|0\rangle_A |i\rangle_I U_2(e_i) |0\rangle_L + |1\rangle_A |i\rangle_I U_1 |0\rangle_L) \\ &= \frac{1}{\sqrt{2N_e}} \sum_{i=1}^{N_e} (|0\rangle_A |i\rangle_I |e_i\rangle_L + |1\rangle_A |i\rangle_I |sp\rangle_L) \end{aligned} \quad (19)$$

Performing the Hadamard gate on the ancilla qubit gives

$$\frac{1}{2\sqrt{N_e}} \sum_{i=1}^{N_e} (|0\rangle_A |i\rangle_I (|e_i\rangle_L + |sp\rangle_L) + |1\rangle_A |i\rangle_I (|e_i\rangle_L - |sp\rangle_L)) \quad (20)$$

Note that the values η_{spe_i} are encoded in the probability amplitudes of the above quantum state Equation (20). For example, the probability of measuring the ancilla qubit and index register being in the quantum state $|0\rangle_A |i\rangle_I$ is given by

$$\Pr(|0\rangle_A |i\rangle_I) = \frac{1}{2N_e} (1 + \Re \langle e_i | sp \rangle_L) = \frac{1}{2N_e} (1 + \eta_{spe_i}) \quad (21)$$

Let us consider an idealistic case for the inference: The negative semantic triples have value functions -1 , while the positive triples have value functions $+1$. In this case, the probability in Equation (21) takes value $\Pr(|0\rangle_A |i\rangle_I) = 0$ if the entity e_i is not a correct return to the query $(s, p, ?)$, while $\Pr(|0\rangle_A |i\rangle_I) = \frac{1}{N_e}$ if the entity e_i is correct.

Since the index register is sampled conditioned on the ancilla qubit, we need to discuss the probability of post-selection on the ancilla qubit. The marginalized probabilities of measuring the ancilla qubit being $|0\rangle_A$ and $|1\rangle_A$ read

$$\begin{aligned} \Pr(|0\rangle_A) &= \frac{1}{2} + \frac{1}{2N_e} \sum_{i=1}^{N_e} \eta_{spe_i} \\ \Pr(|1\rangle_A) &= \frac{1}{2} - \frac{1}{2N_e} \sum_{i=1}^{N_e} \eta_{spe_i} \end{aligned} \quad (22)$$

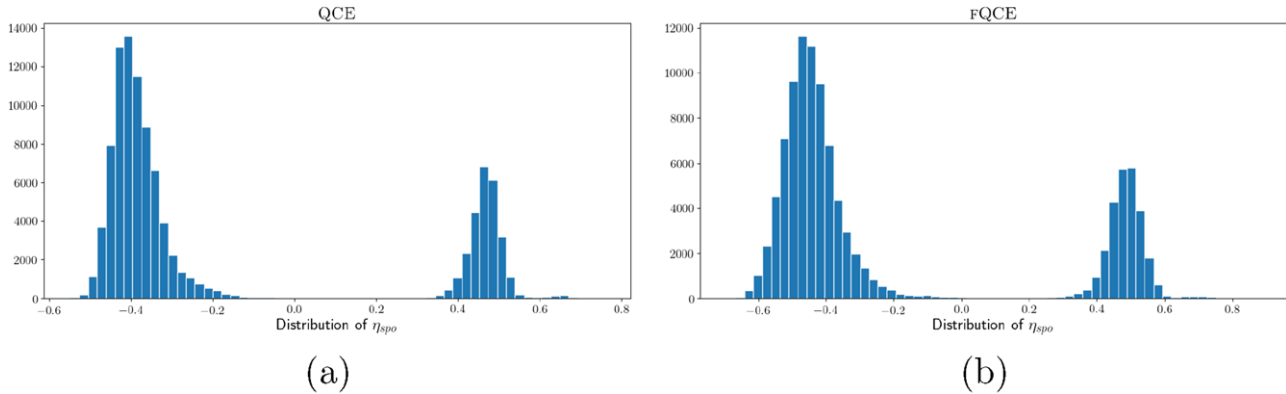


Figure 10. Empirical distributions of the value functions a) η_{spo}^{QCE} and b) η_{spo}^{FQCE} evaluated on the test dataset of KINSHIP. The targets are set as $y_i \in \{-1, 1\}$ during the training. Note that for each triple in the test dataset, say (s, p, o) , the value functions η_{spe_i} and $\eta_{e_i;sp}$, with $i = 1, \dots, N_e$ are evaluated and accumulated for the plotting.

Assume that the cardinality of the solution set to the query $(s, p, ?)$ is $H \in \mathcal{O}(1)$. In the *idealistic* situation, we have the marginalized probability $\Pr(|0\rangle_A) = \frac{H}{2N_e}$, and $\Pr(|1\rangle_A) = 1 - \frac{H}{2N_e}$. To read out the indices that correspond to the entities in the solution set, we can perform amplitude amplification^[27] on the subspace $|0\rangle_A$ of the ancilla qubit. The number of required iterations is approximately $\lfloor \frac{\pi}{4} \sqrt{\frac{2N_e}{H}} \rfloor = \mathcal{O}(\sqrt{N_e})$. The resulting quantum state after the amplitude amplification reads

$$\frac{1}{\sqrt{H}} \sum_{i \in \{i | \phi_p(s, e_i) = 1\}} |0\rangle_A |i\rangle_I \quad (23)$$

It is unnecessary to perform quantum state tomography and read out all the probability amplitudes. We can sample the states of the index register conditioned on $|0\rangle_A$ and determine the most frequent states that are related to the indices of the entities giving the highest scores. Since the cardinality of the solution set is assumed to be $H \in \mathcal{O}(1)$, the same experiment needs to be replicated at least $\mathcal{O}(H\sqrt{N_e})$ times. Hence, this heuristic quantum algorithm realizes a quadratic acceleration with respect to the number of entities N_e .

Our idealistic quantum algorithm provides a quadratic acceleration during the inductive inference on the database. Even a quadratic speedup is desirable when the number of entities N_e is large. Note that another well-known quantum algorithm, Grover's algorithm,^[28] which was designed for searching in a database, also provides a quadratic speedup. More specifically, Grover's algorithm can identify the input to an unknown function in $\mathcal{O}(\sqrt{N})$ steps from a N -item database. At the same time as Grover's publication, it is proved in ref. [29] that Grover's algorithm is an almost optimal solution. Different from this quantum algorithm for the database search, our algorithm is learning-based, adaptive, and inference-oriented.

Note that the above described quantum algorithm is merely *idealistic* and *heuristic*, since the scores of semantic triples in the test dataset take values from the interval $[-1, 1]$ instead of the discrete set $\{-1, 1\}$. Figure 10 shows the empirical distribution of value functions on the KINSHIP test dataset.^[30]

As one can observe that the empirical value functions concentrate around -0.5 and 0.5 . The quantum advantage on inference

might disappear in these cases since $\Pr(|0\rangle_A |i\rangle_I) \approx \Pr(|0\rangle_A |j\rangle_I)$, $\forall i \in \{i | \phi_p(s, e_i) = 1\}$, and $j \notin \{i | \phi_p(s, e_i) = 1\}$. In other words, the probability of sampling correct solutions is approximately equal to the probability of sampling incorrect solutions. Thus, one promising future research direction is to study whether performing nonlinear functions on quantum representations can separate the positive and negative triples in an inference task.

7. Conclusion and Outlook

In this work, we study the quantum Ansätze for the statistical relational learning on knowledge graphs as well as latent quantum representations. Two different quantum models QCE and FQCE are proposed and compared by their complexity and performance. To be specific, QCE assumes that entity representations are stored in a classical data structure, while in the FQCE model quantum entity representations are generated from pure quantum states through unitary circuit evolution. The experiments show that both quantum Ansätze can achieve comparable results to the state-of-the-art classical models on several benchmark datasets.

This work can be further explored in several directions. The quantum circuit architecture could be fine-tuned using reinforcement learning or evolutionary algorithms. It is necessary to understand why quantum circuit models show superior performance on the WN18RR dataset which contains the most entities and the smallest average number of links. Whether this observation indicates that quantum circuit models are only suitable for modeling large but simple relational dataset due to the inherent linearity? Thus, a reasonable question is whether acting nonlinear operations on the quantum representations can improve the inductive inference on complex relational datasets and make the *idealistic* and *heuristic* quantum algorithm for the accelerated inference more realizable?

Appendix A: Preparation of Quantum States

Theorem A1.^[31] Let $\mathbf{x} \in \mathbb{R}^R$ be a real-valued vector. The quantum state $|x\rangle = \frac{1}{\|\mathbf{x}\|_2} \sum_{i=1}^R x_i |i\rangle$ can be prepared using $\lceil \log_2 R \rceil$ qubits in time $\mathcal{O}(\log_2 R)$.

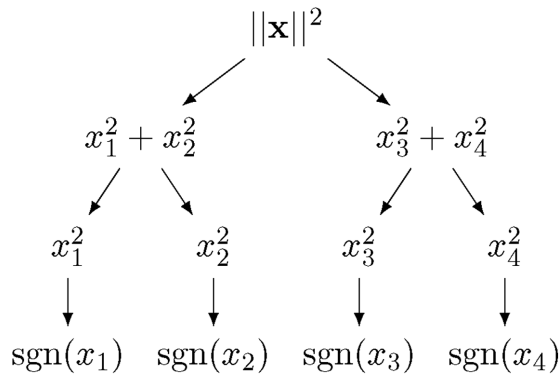


Figure A1. Classical memory structure with quantum access for creating the quantum state $|x\rangle = x_1 |00\rangle + x_2 |01\rangle + x_3 |10\rangle + x_4 |11\rangle$.

Theorem A1 claims that there exist a classical memory structure and a quantum algorithm which can load classical data into a quantum state with exponential acceleration. **Figure A1** illustrates a simple example. Given an $R = 4D$ real-valued vector, the quantum state $|x\rangle = x_1 |00\rangle + x_2 |01\rangle + x_3 |10\rangle + x_4 |11\rangle$ can be prepared by querying the classical memory structure and applying three controlled rotations.

Let us assume that x is normalized, namely $\|x\|_2 = 1$. The quantum state $|x\rangle$ is created from the initial state $|0\rangle |0\rangle$ by querying the memory structure from the root to the leaf. The first rotation is applied on the first qubit, giving

$$(\cos \theta_1 |0\rangle + \sin \theta_1 |1\rangle) |0\rangle = \left(\sqrt{x_1^2 + x_2^2} |0\rangle + \sqrt{x_3^2 + x_4^2} |1\rangle \right) |0\rangle$$

where $\theta_1 := \tan^{-1} \sqrt{\frac{x_3^2 + x_4^2}{x_1^2 + x_2^2}}$. The second rotation is applied on the second qubit conditioned on the state of qubit 1. It gives

$$\begin{aligned} & \sqrt{x_1^2 + x_2^2} |0\rangle \frac{1}{\sqrt{x_1^2 + x_2^2}} (|x_1| |0\rangle + |x_2| |1\rangle) + \\ & \sqrt{x_3^2 + x_4^2} |1\rangle \frac{1}{\sqrt{x_3^2 + x_4^2}} (|x_3| |0\rangle + |x_4| |1\rangle) \end{aligned}$$

The last rotation loads the signs of coefficients conditioned on qubits 1 and 2. In general, an R -dimensional real-valued vector needs to be stored in a classical memory structure with $\lceil \log_2 R \rceil + 1$ layers. The data vector can be loaded into a quantum state using $\mathcal{O}(\log_2 R)$ nontrivial controlled rotations.

Acknowledgements

This research was supported by the BMBF funded project Machine Learning with Knowledge Graphs, and Siemens Corporate Technology.

Conflict of Interest

The authors declare no conflict of interest.

Keywords

inference on relational database, knowledge graphs, quantum acceleration, representation learning, variational quantum circuit

Received: September 5, 2018

Revised: December 12, 2018

Published online:

- [1] M. Nickel, V. Tresp, H. P. Kriegel, presented at *Proc. of the 28th Int. Conf. on Machine Learning*, Bellevue, WA, June 28–July 02, **2011**.
- [2] B. Yang, W. t. Yih, X. He, J. Gao, L. Deng, *arXiv preprint arXiv:1412.6575*, **2014**.
- [3] T. Trouillon, J. Welbl, S. Riedel, É. Gaussier, G. Bouchard, in *Proc. of the 33rd Int. Conf. on Machine Learning (ICML 2016)*, New York, **2016**, pp. 2071–2080.
- [4] M. Nickel, L. Rosasco, T. A. Poggio, in *Proc. of the 30th AAAI Conf. on Artificial Intelligence*, Phoenix, AZ, **2016**, pp. 1955–1961.
- [5] L. R. Tucker, *Psychometrika* **1966**, 31, 279.
- [6] M. Nickel, V. Tresp, in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, Springer, **2013**, pp. 272–287.
- [7] M. Schuld, A. Bocharov, K. Svore, N. Wiebe, *arXiv preprint arXiv:1804.00633*, **2018**.
- [8] I. Kerenidis, A. Prakash, *arXiv preprint arXiv:1603.08675*, **2016**.
- [9] H. Buhrman, R. Cleve, J. Watrous, R. De Wolf, *Phys. Rev. Lett.* **2001**, 87, 167902.
- [10] J. C. Garcia-Escartin, P. Chamorro-Posada, *Phys. Rev. A* **2013**, 87, 052330.
- [11] P. Chamorro-Posada, J. C. Garcia-Escartin, *arXiv preprint arXiv:1706.06564*, **2017**.
- [12] A. M. Childs, N. Wiebe, *arXiv preprint arXiv:1202.5822*, **2012**.
- [13] A. Asuncion, D. Newman, UCI Machine Learning Repository, **2007**.
- [14] K. Toutanova, D. Chen, in *Proc. of the 3rd Workshop on Continuous Vector Space Models and Their Compositionality*, Association for Computational Linguistics, **2015**, pp. 57–66.
- [15] T. Dettmers, P. Minervini, P. Stenetorp, S. Riedel, *arXiv preprint arXiv:1707.01476*, **2017**.
- [16] K. Leetaru, P. A. Schrod, presented at *ISA Annual Convention*, San Francisco, CA, April 3–6, **2013**.
- [17] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, O. Yakhnenko, *Advances in Neural Information Processing Systems 26 (NIPS 2013)*, **2013**, pp. 2787–2795.
- [18] By the time of finishing this project, none of the quantum computing cloud platforms provide fully tunable entangled qubits.
- [19] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D. G. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu, X. Zheng, *Proc. of 12th USENIX Conf. on Operating Systems Design and Implementation (OSDI '16)*, **2016**, pp. 265–283.
- [20] Y. Ma, M. Hildebrandt, S. Baier, V. Tresp, Holistic Representations for Memorization and Inference, *Proc. of 34th Conf. on Uncertainty in Artificial Intelligence (UAI 2018)*, Monterey, CA, **2018**, pp. 403–413.
- [21] Note that *best known* models might employ arbitrarily complex structures, for example, convolutional or recurrent neural networks. For comparison, we provide the number of trainable parameters in different models. Note that the number of trainable parameters depends not only on the model structure but also the number of different entities and predicates in the dataset. In the case of the FB15k-237 dataset, the state-of-the-art model described in ref. [15] contains

- 5.05 parameters, and the RESCAL model contains 1.89M parameters, while the fQCE model contains 1.06M parameters.
- [22] M. Schuld, I. Sinayskiy, F. Petruccione, *Quantum Inf. Process.* **2014**, *13*, 2567.
- [23] E. Torrontegui, J. J. Garcia-Ripoll, *arXiv preprint arXiv:1801.00934*, **2018**.
- [24] G. Huang, Y. Sun, Z. Liu, D. Sedra, K. Q. Weinberger, in *European Conference on Computer Vision*, Springer, **2016**, pp. 646–661.
- [25] S. Hochreiter, J. Schmidhuber, *Neural Comput.* **1997**, *9*, 1.
- [26] L. van der Maaten, G. Hinton, *J. Mach. Learn. Res.* **2008**, *9*, 2579.
- [27] G. Brassard, P. Hoyer, M. Mosca, A. Tapp, *Contemp. Math.* **2002**, *305*, 53.
- [28] L. K. Grover, in *Proceedings of the 28th Annual ACM Symp. on Theory of Computing*, ACM, **1996**, pp. 212–219.
- [29] C. H. Bennett, E. Bernstein, G. Brassard, U. Vazirani, *SIAM J. Comput.* **1997**, *26*, 1510.
- [30] The empirical distributions are obtained in the following way: Given a semantic triple (s, p, o) in the test dataset, we calculate the value functions η_{spe_i} and η_{epo_i} , with $i = 1, \dots, N_e$.
- [31] A. Prakash, *Ph.D. Thesis*, UC Berkeley, **2014**.

Chapter 5

Quantum Machine Learning Algorithm for Knowledge Graphs

Quantum Machine Learning Algorithm for Knowledge Graphs

YUNPU MA, Ludwig Maximilian University of Munich & Siemens CT

YUYI WANG, ETHZ

VOLKER TRESP, Ludwig Maximilian University of Munich & Siemens CT

Semantic knowledge graphs are large-scale triple-oriented databases for knowledge representation and reasoning. Implicit knowledge can be inferred by modeling and reconstructing the tensor representations generated from knowledge graphs. However, as the sizes of knowledge graphs continue to grow, classical modeling becomes increasingly computational resource intensive. This paper investigates how quantum resources can be capitalized to accelerate the modeling of knowledge graphs. In particular, we propose the first quantum machine learning algorithm for making inference on tensorized data, e.g., on knowledge graphs. Since most tensor problems are NP-hard Hillar and Lim [16], it is challenging to devise quantum algorithms to support that task. We simplify the problem by making a plausible assumption that the tensor representation of a knowledge graph can be approximated by its low-rank tensor singular value decomposition, which is verified by our experiments. The proposed sampling-based quantum algorithm achieves exponential speedup with a runtime that is polylogarithmic in the dimension of knowledge graph tensor.

CCS Concepts: • **Computing methodologies** → *Knowledge representation and reasoning; Statistical relational learning.*

Additional Key Words and Phrases: knowledge graphs, relational database, quantum tensor singular value decomposition, quantum machine learning

ACM Reference Format:

Yunpu Ma, Yuyi Wang, and Volker Tresp. 2018. Quantum Machine Learning Algorithm for Knowledge Graphs. 1, 1 (January 2018), 24 pages. <https://doi.org/10.1145/1122445.1122456>

1 INTRODUCTION

Semantic knowledge graphs (KGs) are graph-structured databases consisting of semantic triples (*subject, predicate, object*), where subject and object are nodes in the graph, and the predicate is the label of a directed link between subject and object. An existing triple normally represents a fact, e.g., (*California, located_in, USA*) and missing triples stand for triples known to be false (closed-world assumption) or with an unknown truth value. In recent years a number of sizable knowledge graphs have been built, such as FREEBASE [3], YAGO [30], etc. The largest knowledge graph, e.g., Google’s Knowledge Vault [8], contains more than 100 billion facts and hundreds of millions of distinguishable entities.

An adjacency tensor can represent a knowledge graph with three dimensions: One stands for subjects, one for predicates and one for objects. More precisely, we let $\chi \in \{0, 1\}^{d_1 \times d_2 \times d_3}$ denote the semantic tensor of a knowledge graph, where d_1 , d_2 , and d_3 represent the number of subjects, predicates, and objects, respectively. An entry χ_{spo} in χ assumes the value 1 if the semantic triple (s, p, o) is known to be true, while it assumes the value 0 if the triple is false or

Authors’ addresses: Yunpu Ma, cognitive.yunpu@gmail.com, Ludwig Maximilian University of Munich & Siemens CT; Yuyi Wang, ETHZ; Volker Tresp, volker.tresp@siemens.com, Ludwig Maximilian University of Munich & Siemens CT.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2018 Association for Computing Machinery.

Manuscript submitted to ACM

missing. A goal of machine learning is to infer the truth value of triples, given the knowledge graph triples were known to be true. Popular learning-based algorithms for modeling KGs are based on a factorization of the adjacency tensor, e.g., the Tucker tensor decomposition, PARAFAC, RESCAL [24], or compositional models, e.g., DistMult [36], and HolE [26].

The vast number of facts and entities makes it particularly challenging to scale learning and inference algorithms to perform inference on the entire knowledge graph. The goal of this paper is to use quantum computation to design algorithms that can dramatically accelerate the inference task. Thanks to the rapid development of quantum computing technologies, quantum machine learning [2] is becoming an active research area which attracts researchers from different communities. In general, quantum machine learning exhibits great potential for accelerating classical algorithms, e.g., solving linear systems of equations [15], supervised and unsupervised learning [35], support vector machines [28], Gaussian processes [6], non-negative matrix factorization [10], recommendation systems [17], etc.

Note that most of the above-mentioned quantum machine learning algorithms contain subroutines for singular value decomposition, singular value estimation, and singular value projection of data matrices that are prepared and presented as quantum density matrices. We show that the tensor factorization algorithm presented in this paper, which uses existing quantum algorithms as subroutines, has a polylogarithmic runtime complexity. However, unlike matrices, most tensor problems are NP-hard, and there is no current quantum algorithm which can handle tensorized data. Therefore, to understand the difficulties of designing quantum machine learning algorithms on tensorized data, e.g., data derived from a vast relational database, we need first to answer the following questions:

(1) Under what conditions can we infer implicit knowledge from an incomplete knowledge graph by reconstructing it via classical algorithms; (2) Does there exist an analogous tensor singular value decomposition method that we can map to a quantum algorithm? (3) Assuming that the knowledge graph has global and well-defined relational patterns, can the tensor SVD of a subsampled semantic tensor well approximate the original tensor. Mainly, after projecting onto the lower-rank space, previously unobserved truth values of semantic triples might be boosted? (4) If all the above conditions are fulfilled, how can we design a quantum algorithm which projects the tensorized data onto lower-rank space to reconstruct the original tensor?

The first part of this paper contributes to the classical theory of binary tensor sparsification. As a novel contribution, we derive the first binary tensor sparsification condition under which the original tensor can be well approximated by the truncated or projected tensor SVD of its subsampled tensor. The second part focuses on developing the quantum machine learning algorithm. To handle the tensorized data, we first explain a quantum tensor contraction subroutine. We then design a quantum learning algorithm on knowledge graphs using quantum principal component analysis, quantum phase estimation, and quantum singular value projection. We study the runtime complexity and show that this sampling-based quantum algorithm provides exponential acceleration w.r.t. the size of the knowledge graph during inference.

1.1 Related Work

In this section, we discuss recent work on quantum machine learning for big data. It is commonly believed that the quantum recommendation system (QRS) proposed in Kerenidis and Prakash [17] will potentially be one of the first commercial applications of quantum machine learning. The quantum recommendation system provides personalized recommendations to individual users according to a preference matrix A with runtime $O(\text{poly}(k)\text{polylog}(mn))$, where $m \times n$ is the size of the preference matrix A which is assumed to have a low rank- k approximation. On the other hand, a recent breakthrough made by Tang [31] shows that by dequantizing the quantum recommendation algorithm a classical machine learning algorithm can achieve the same acceleration if the classical algorithm has access to a data structure

which resembles the one required in the QRS. However, as commented by the authors of Kerenidis and Prakash [17] in Kerenidis et al. [18], this new classical algorithm based on the FKV methods Frieze et al. [12] has a much worse polynomial dependence on the rank of the preference matrix and a dramatic slowdown dependence on a predefined precision parameter, making it completely impractical. Therefore, it remains an open question to find the corresponding dequantized classical algorithms for machine learning on tensorized data that are polylogarithmic in dimension as the proposed quantum algorithm.

A recent work Gu et al. [14] that presents a quantum algorithm for higher-order tensor singular value decomposition (HOSVD) De Lathauwer et al. [7]. The quantum HOSVD algorithm decomposes a m -way n -dimensional tensor into a core tensor and unitary matrices with computational complexity $O(mn^{3/2} \log^m n)$. It provides an exponential acceleration compared with the classical HOSVD with complexity $O(mn^{m+1})$. Note that the polynomial dependence of the complexity on the tensor dimension comes from the quantum subroutines since the quantum HOSVD reconstruct the core tensor and unitary matrices explicitly. In contrast, our quantum tensor SVD method doesn't estimate singular values and unitary matrices explicitly, instead, it samples results from a projected tensor under the assumption that the tensorized data has a low-rank orthogonal approximation. Hence, it provides a polylogarithmic dependence on the tensor dimension.

2 TENSOR SINGULAR VALUE DECOMPOSITION

First, we recap the singular value decomposition (SVD) of matrices. Then we introduce tensor SVD and show that a given tensor can be reconstructed with a small error from the low-rank tensor SVD of the subsampled tensor. Other tensor decomposition algorithms, e.g., higher-order tensor SVD De Lathauwer et al. [7], will not be considered in this work since designing their quantum counterparts can be much more involved.

SVD Let $A \in \mathbb{R}^{m \times n}$, the SVD is a factorization of A is the form $A = U\Sigma V^T$, where Σ is a rectangle diagonal matrix singular values on the diagonal, $U \in \mathbb{R}^{m \times m}$ and $V \in \mathbb{R}^{n \times n}$ are orthogonal matrices with $U^T U = U U^T = I_m$ and $V^T V = V V^T = I_n$.

Notations for Tensors A N -way tensor is defined as $\mathcal{A} = (\mathcal{A}_{i_1 i_2 \dots i_N}) \in \mathbb{R}^{d_1 \times d_2 \times \dots \times d_N}$, where d_k is the k -th dimension. Given two tensors \mathcal{A} and \mathcal{B} with the same dimensions, the inner product is defined as $\langle \mathcal{A}, \mathcal{B} \rangle_F := \sum_{i_1=1}^{d_1} \dots \sum_{i_N=1}^{d_N} \mathcal{A}_{i_1 i_2 \dots i_N} \mathcal{B}_{i_1 i_2 \dots i_N}$. The Frobenius norm is defined as $\|\mathcal{A}\|_F := \sqrt{\langle \mathcal{A}, \mathcal{A} \rangle_F}$. The spectral norm $\|\mathcal{A}\|_\sigma$ of the tensor \mathcal{A} is defined as $\|\mathcal{A}\|_\sigma = \max\{\|\mathcal{A} \otimes_1 \mathbf{x}_1 \dots \otimes_N \mathbf{x}_N\|_F \mid \mathbf{x}_k \in S^{d_k-1}, k = 1, \dots, N\}$, where the tensor-vector product is defined as

$$\mathcal{A} \otimes_1 \mathbf{x}_1 \dots \otimes_N \mathbf{x}_N := \sum_{i_1=1}^{d_1} \dots \sum_{i_N=1}^{d_N} \mathcal{A}_{i_1 i_2 \dots i_N} x_{1i_1} x_{2i_2} \dots x_{Ni_N}$$

and S^{d_k-1} denotes the unit sphere in \mathbb{R}^{d_k} .

Tensor SVD Parallel to the matrix singular value decomposition, several orthogonal tensor decompositions with different definitions of orthogonality are studied in Kolda [20]. Among them the *complete orthogonal rank decomposition* is also referred to as the *tensor singular value decomposition* (tensor SVD, c.f. Definition 1) studied in Chen and Saad [5]. Especially, Zhang and Golub [37] shows that for all tensors with $N \geq 3$, the tensor SVD can be uniquely determined via incremental rank-1 approximation.

DEFINITION 1. If a tensor $\mathcal{A} \in \mathbb{R}^{d_1 \times d_2 \times \dots \times d_N}$ can be written as sum of rank-1 outer product tensors $\mathcal{A} = \sum_{i=1}^R \sigma_i u_1^{(i)} \otimes u_2^{(i)} \dots \otimes u_N^{(i)}$, with singular values $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_R$ and $\langle u_k^{(i)}, u_k^{(j)} \rangle = \delta_{ij}$ for $k = 1, \dots, N$. Then \mathcal{A} has a tensor singular value decomposition with rank R .

Define the orthogonal matrices $U_k = [u_k^{(1)}, u_k^{(2)}, \dots, u_k^{(R)}] \in \mathbb{R}^{d_k \times R}$ with $U_k^T U_k = \mathbb{I}_R$ for $k = 1, \dots, N$, and the diagonal tensor $\mathcal{D} \in \mathbb{R}^{R \times R \times \dots \times R}$ with $\mathcal{D}_{ii \dots i} = \sigma_i$, then the tensor SVD for \mathcal{A} can be also written as $\mathcal{A} = \mathcal{D} \otimes_1 U_1 \otimes_2 U_2 \dots \otimes_N U_N$. Given an arbitrary tensor $\mathcal{A} \in \mathbb{R}^{d_1 \times d_2 \times \dots \times d_N}$, an interesting question is to find a low-rank approximation via tensor SVD. In particular, Chen and Saad [5] proves the existence of the global optima of the following optimization problem

$$\min \|\mathcal{A} - \sum_{i=1}^r \sigma_i u_1^{(i)} \otimes u_2^{(i)} \dots \otimes u_N^{(i)}\|_F ; \text{ s.t. } \langle u_k^{(i)}, u_k^{(j)} \rangle = \delta_{ij}, \text{ for } k = 1, \dots, N$$

for any $r \leq \min\{d_1, d_2, \dots, d_N\}$. We will utilize this fact to derive the error bound after projecting the tensor onto low-rank subspaces. Note that, in contrast to the matrix SVD, tensor SVD is unique up to the signs of singular values.

Our quantum algorithm builds on the assumption that the semantic tensor χ can be well approximated by a low-rank tensor $\hat{\chi}$ with $\|\chi - \hat{\chi}\|_F \leq \epsilon \|\chi\|_F$ for small $\epsilon > 0$. Previous work of recommendation systems Drineas et al. [9] has shown that the quality of recommendations for users depends on the reconstruction error. Similarly, in the case of relational learning, with a bounded tensor approximation error it is possible to estimate the probability of a *successful* information retrieval. Consider the query (s, p, ?) on a KG using classical algorithm. We normally only readout top- n returns from the reconstructed tensor $\hat{\chi}$, written as $\hat{x}_{sp1}, \dots, \hat{x}_{spn}$, where n is a small integer corresponding to the commonly used Hits@ n metric. The information retrieval is called *successful* if the correct object corresponding to the query can be found in the returned list $\hat{x}_{sp1}, \dots, \hat{x}_{spn}$. In particular, we have the following estimation.

LEMMA 1. *If an algorithm returns an approximation of the binary semantic tensor χ , denoted $\hat{\chi}$, with $\|\chi - \hat{\chi}\|_F \leq \epsilon \|\chi\|_F$ and $\epsilon < \frac{1}{2}$, then the probability of a successful information retrieval from the top- n returns of $\hat{\chi}$ is at least $1 - (\frac{\epsilon}{1-\epsilon})^n$. (Proof in Appendix A.1)*

In real-world applications, we can only observe part of the non-zero entries in a given tensor \mathcal{A} , and the task is to infer unobserved non-zero entries with high probability. This task corresponds to items recommendation for users given an observed preference matrix, or implicit knowledge inference given partially observed relational data. The partially observed tensor is called as subsampled or sparsified, denoted $\hat{\mathcal{A}}$. Without further specifying the dimensionality of the tensor, we consider the following subsampling and rescaling scheme proposed in Achlioptas and McSherry [1]:

$$\hat{\mathcal{A}}_{i_1 i_2 \dots i_N} = \begin{cases} \frac{\mathcal{A}_{i_1 i_2 \dots i_N}}{p} & \text{with probability } p \\ 0 & \text{otherwise.} \end{cases} \quad (1)$$

It means that the non-zero elements of a tensor are independently and identically sampled with the probability p and rescaled afterwards. The subsampled tensor can be rewritten as $\hat{\mathcal{A}} = \mathcal{A} + \mathcal{N}$, where \mathcal{N} is a noise tensor. Entries of \mathcal{N} are independent random variables with distribution $\Pr(\mathcal{N}_{i_1 \dots i_N} = (1/p - 1)\mathcal{A}_{i_1 \dots i_N}) = p$ and $\Pr(\mathcal{N}_{i_1 \dots i_N} = -\mathcal{A}_{i_1 \dots i_N}) = 1 - p$.

Now, the task is to reconstruct the original tensor \mathcal{A} by modeling $\hat{\mathcal{A}}$. We use tensor SVD to model the observed tensor $\hat{\mathcal{A}}$. The reconstruction error can be bounded either using the truncated r -rank tensor SVD, denoted $\hat{\mathcal{A}}_r$, or the projected tensor SVD with absolute singular value threshold τ , denoted $\hat{\mathcal{A}}_{|\cdot| \geq \tau}$. Notation $\hat{\mathcal{A}}_{|\cdot| \geq \tau}$ means that the subsampled tensor $\hat{\mathcal{A}}$ is projected onto the eigenspaces with absolute singular values larger than a cutoff threshold $\tau > 0$. By comparison, in matrix SVD, essentially the singular values larger than, or equal to, a cutoff threshold are kept and those that are smaller are disregarded. However, in the tensor case, negative singular values can arise. The same

cutoff scheme then is no longer meaningful, as it would disregard singular values with large negative values which may potentially be important.

Theorem 1 gives the reconstruction error bound using \mathcal{A}_r and the corresponding conditions on the sample probability.

THEOREM 1. *Let $\mathcal{A} \in \{0, 1\}^{d_1 \times d_2 \times \dots \times d_N}$. Suppose that \mathcal{A} can be well approximated by its r -rank tensor SVD \mathcal{A}_r . Using the subsampling scheme defined in Eq. 1 with the sample probability $p \geq \max\{0.22, 8r \left(\log(\frac{2N}{N_0}) \sum_{k=1}^N d_k + \log \frac{2}{\delta} \right) / (\tilde{\epsilon} \|\mathcal{A}\|_F)^2\}$, $N_0 = \log \frac{3}{2}$, then the original tensor \mathcal{A} can be reconstructed from the truncated tensor SVD of the subsampled tensor $\hat{\mathcal{A}}$. The error satisfies $\|\mathcal{A} - \hat{\mathcal{A}}_r\|_F \leq \epsilon \|\mathcal{A}\|_F$ with probability at least $1 - \delta$, where ϵ is a function of $\tilde{\epsilon}$. Especially, $\tilde{\epsilon}$ together with the sample probability controls the norm of the noise tensor.*

PROOF. We outline the ideas involved in the proof and relegate details to the appendix A.2. The proof is divided into two parts. We first derive the following bound for the reconstruction error (see appendix Lemma A 2, 3, 4)

$$\|\mathcal{A} - \hat{\mathcal{A}}_r\|_F \leq 2\|\mathcal{A} - \mathcal{A}_r\|_F + 2\sqrt{\|\mathcal{A}_r\|_F \|\mathcal{A} - \mathcal{A}_r\|_F} + 2\sqrt{\|\mathcal{N}_r\|_F \|\mathcal{A}_r\|_F} + \|\mathcal{N}_r\|_F.$$

Notice that the RHS doesn't contain the subsampled tensor $\hat{\mathcal{A}}$. Therefore we can further simplify the RHS by assuming that the original tensor has a low-rank approximation, namely $\|\mathcal{A} - \mathcal{A}_r\|_F \leq \epsilon_0 \|\mathcal{A}\|_F$. After that, we prove numerically that the random variables $\mathcal{N}_{i_1 \dots i_N} x_{1i_1} \dots x_{Ni_N}$ for any $\mathbf{x}_k \in S^{d_k-1}$, $k = 1, \dots, N$ are sub-Gaussian distributed if the sample probability fulfills $p \gtrsim 0.22$. Hence we can further use the covering number on the product space $S^{d_1-1} \times \dots \times S^{d_N-1}$ to bound the norm of \mathcal{N} (see appendix Lemma A 5, 6, 7):

$$\|\mathcal{N}_r\|_F \leq \sqrt{r \frac{8}{p} \left(\log(\frac{2N}{N_0}) \sum_{k=1}^N d_k + \log \frac{2}{\delta} \right)}. \quad (2)$$

Finally, by requiring $\|\mathcal{N}_r\|_F \leq \tilde{\epsilon} \|\mathcal{A}\|_F$ or by selecting

$$p \geq \max\{0.22, 8r \left(\log(\frac{2N}{N_0}) \sum_{k=1}^N d_k + \log \frac{2}{\delta} \right) / (\tilde{\epsilon} \|\mathcal{A}\|_F)^2\}$$

we have $\|\mathcal{A} - \hat{\mathcal{A}}_r\|_F \leq \epsilon \|\mathcal{A}\|_F$ via Eq. 2, where $\epsilon := 2(\epsilon_0 + \sqrt{\epsilon_0} + \sqrt{\tilde{\epsilon}}) + \tilde{\epsilon}$. ■

We further introduce the projected tensor SVD $\hat{\mathcal{A}}_{|\cdot| \geq \tau}$ and analysis its error bound for the later use in the quantum singular value projection. Note that quantum algorithms are fundamentally different from classical algorithms. For example, classical algorithms for matrix factorization approximate a low-rank matrix by projecting it onto a subspace spanned by the eigenspaces possessing top- r singular values with predefined small r . Quantum subroutine, e.g., quantum singular value estimation, on the other hand, can read and store all singular values of a unitary operator into a quantum register. However, singular values stored in the quantum register cannot be read out and compared simultaneously since quantum state collapses after one measurement; measuring the singular values one by one will also break the quantum advantage. Therefore, we perform a projection onto the union of operator's subspaces whose singular values are larger than a threshold; and this step can be implemented on the quantum register without destroying the superposition. Moreover, since we use quantum PCA as a subroutine which ignores the sign of singular values during the projection, we have to analyze the reconstruction error given by $\hat{\mathcal{A}}_{|\cdot| \geq \tau}$ for the quantum algorithm. Theorem 2 gives the reconstruction error bound using $\hat{\mathcal{A}}_{|\cdot| \geq \tau}$ and conditions for the sample probability.

THEOREM 2. *Let $\mathcal{A} \in \{0, 1\}^{d_1 \times d_2 \times \dots \times d_N}$. Suppose that \mathcal{A} can be well approximated by its r -rank tensor SVD \mathcal{A}_r . Using the subsampling scheme defined in Eq. 1 with the sample probability $p \geq \max\{0.22, p_1 := \frac{l_1 C_0}{(\tilde{\epsilon} \|\mathcal{A}\|_F)^2}, p_2 := \frac{r C_0}{(\tilde{\epsilon} \|\mathcal{A}\|_F)^2}, p_3 :=$*

$\frac{\sqrt{2rC_0}}{\epsilon_1 \tilde{\epsilon} \|\mathcal{A}\|_F}$), with $C_0 = 8 \left(\log\left(\frac{2N}{N_0}\right) \sum_{k=1}^N d_k + \log \frac{2}{\delta} \right)$, $N_0 = \log \frac{3}{2}$, where l_1 denotes the largest index of singular values of tensor $\hat{\mathcal{A}}$ with $\sigma_{l_1} \geq \tau$, and choosing the threshold as $0 < \tau \leq \frac{\sqrt{2C_0}}{p\tilde{\epsilon}}$, then the original tensor \mathcal{A} can be reconstructed from the projected tensor SVD of $\hat{\mathcal{A}}$. The error satisfies $\|\mathcal{A} - \hat{\mathcal{A}}_{|\cdot| \geq \tau}\|_F \leq \epsilon \|\mathcal{A}\|_F$ with probability at least $1 - \delta$, where ϵ is a function of $\tilde{\epsilon}$ and ϵ_1 . Especially, $\tilde{\epsilon}$ together with p_1 and p_2 determine the norm of noise tensor and ϵ_1 together with p_3 control the value of $\hat{\mathcal{A}}$'s singular values that are located outside the projection boundary.

PROOF. The proof resembles that of Theorem 1, and details are relegated in appendix A.2. One can first derive $\|\mathcal{A} - \hat{\mathcal{A}}_{|\cdot| \geq \tau}\|_F \leq 3\|\mathcal{A} - \hat{\mathcal{A}}_{l_1}\|_F$. Then we distinguish two cases: $l_1 \geq r$ and $l_1 < r$ and show that if $p \geq \max\{0.22, p_1, p_2\}$ it gives $\|\mathcal{N}_r\|_F \leq \tilde{\epsilon} \|\mathcal{A}\|_F$ via Eq. 2. Moreover, requiring $p \geq p_3$ leads to $\|\hat{\mathcal{A}}_r - \hat{\mathcal{A}}_{l_1}\|_F \leq \epsilon_1 \|\mathcal{A}\|_F$. It says that the singular values of $\hat{\mathcal{A}}$ that are outside the projection boundary can be controlled by p_3 and predefined small ϵ_1 . Notice that $p_3 \gg p_1, p_2$ if tensor \mathcal{A} is dense and $\|\mathcal{A}\|_F$ is large enough. Hence we can estimate sample probability $p \geq \{0.22, p_3\}$ given predefined $\tilde{\epsilon}, \epsilon_1$ without knowing l_1 a priori. On the other hand, this theorem indicates that it is impossible to complete an over sparsified tensor with subsample probability smaller than 0.22. ■

In the bodies of Theorem 1 and 2 there exist data-dependent parameters r and l_1 which are unknown a priori. These parameters can only be estimated by performing tensor SVD to the original and subsampled tensors explicitly. However, in practice, mostly, we are only given the subsampled tensor without even knowing the subsample probability. For example, given an incomplete semantic tensor, we do not know what percentage of information is missing, and therefore we cannot rescale the entries in the incomplete tensor. Fortunately, unlike any other matrix sparsification Achlioptas and McSherry [1] or tensor sparsification algorithms Nguyen et al. [23], our analysis suggests a reasonable initial guess for the subsample probability numerically, and inversely an initial guess for the lower-rank r and the projection threshold τ as well.

3 QUANTUM MACHINE LEARNING ALGORITHM FOR KNOWLEDGE GRAPHS

3.1 Quantum Mechanics

To make this work self-consistent we briefly introduce the Dirac notations of quantum mechanics. Under Dirac's convention quantum states can be represented as complex-valued vectors in a Hilbert space \mathcal{H} . For example, a two-dimensional complex Hilbert \mathcal{H}_2 space can describe the quantum state of a spin-1 particle, which provides the physical realization of a qubit. By default, the basis in \mathcal{H}_2 for a spin-1 qubit read $|0\rangle = [1, 0]^T$ and $|1\rangle = [0, 1]^T$. The Hilbert space of a n -qubits system has dimension 2^n whose computational basis can be chosen as the canonical basis $|i\rangle \in \{|0\rangle, |1\rangle\}^{\otimes n}$, where \otimes represents tensor product. Hence any quantum state $|\phi\rangle \in \mathcal{H}_{2^n}$ can be written as a quantum superposition $|\phi\rangle = \sum_{i=1}^{2^n} \phi_i |i\rangle$, where the coefficients $|\phi_i|^2$ can also be interpreted as the probability of observing the canonical basis state $|i\rangle$ after measuring $|\phi\rangle$ using canonical basis. Moreover, we use $\langle\phi|$ to represent the conjugate transpose of $|\phi\rangle$, i.e., $(|\phi\rangle)^\dagger = \langle\phi|$. Given two states $|\phi\rangle$ and $|\psi\rangle$ the inner product on the Hilbert space is defined as $\langle\phi|\psi\rangle^* = \langle\psi|\phi\rangle$. A density matrix is a projection operator which is used to describe the statistics of a quantum system. For example, the density operator of the mixed state $|\phi\rangle$ in the canonical basis reads $\rho = \sum_{i=1}^{2^n} |\phi_i|^2 |i\rangle\langle i|$. Moreover, given two subsystems with density matrices ρ and σ the density matrix for the whole system is their tensor product, namely $\rho \otimes \sigma$.

The time evolution of a quantum state is generated by the Hamiltonian of the system. The Hamiltonian H is a Hermitian operator with $H^\dagger = H$. Let $|\phi(t)\rangle$ denote the quantum state at time t under the evolution of an invariant Hamiltonian H . Then according to the Schrödinger equation $|\phi(t)\rangle = e^{-iHt} |\phi(0)\rangle$, where the unitary operator e^{-iHt} can be written as the matrix exponentiation of the Hermitian matrix H , i.e., $e^{-iHt} = \sum_{n=0}^{\infty} \frac{(-iHt)^n}{n!}$. Eigenvectors of the

Hamiltonian H , denoted $|u_i\rangle$, also form a basis of the Hilbert space. Then the spectral decomposition of the Hamiltonian H reads $H = \sum_i \lambda_i |u_i\rangle \langle u_i|$, where λ_i is the eigenvalue or the energy level of the system. Therefore, the evolution operator of a time-invariant Hamiltonian can be rewritten as

$$e^{-iHt} = e^{-it \sum_i \lambda_i |u_i\rangle \langle u_i|} = \sum_i e^{-i\lambda_i t} |u_i\rangle \langle u_i|, \quad (3)$$

where we use the observation $(|u_i\rangle \langle u_i|)^n = |u_i\rangle \langle u_i|$ for $n = 1, \dots, \infty$. When applying it on an arbitrary initial state $|\phi(0)\rangle$ we obtain $|\phi(t)\rangle = e^{-iHt} |\phi(0)\rangle = \sum_i e^{-i\lambda_i t} \beta_i |u_i\rangle$, where β_i indicates the overlap between the initial state and the eigenbasis of H , i.e., $\beta_i := \langle u_i | \phi(0) \rangle$. To implement the time evolution operator e^{-iHt} and simulate the dynamics of a quantum system using universal quantum circuits is a challenging task since it involves the matrix exponentiation of a possibly dense matrix. Therefore, Hamiltonian simulation is an active research area which was first proposed by Richard Feynman Feynman [11], see also Lloyd [21].

3.2 Quantum Tensor Singular Value Decomposition

In this section, we propose a quantum algorithm for inference on knowledge graphs using quantum singular value estimation and projection. In the following, a 3-dimensional semantic tensor $\chi \in \{0, 1\}^{d_1 \times d_2 \times d_3}$ as one example of a tensor \mathcal{A} is of particular interest. The present method builds on the assumption that the original semantic tensor χ modeling the complete knowledge graph has a low-rank orthogonal approximation, denoted χ_r , with small rank r . The low-rank assumption is plausible if the knowledge graph contains global and well-defined relational patterns, as has been discussed in Nickel et al. [25]. χ could be thereof reconstructed approximately from $\hat{\chi}$ via tensor SVD according to Theorem 1 and 2. Since our quantum algorithm is sampling-based instead of learning-based, w.l.o.g., we consider sampling the correct objects given the query $(s, p, ?)$ as an example and discuss the runtime complexity of one inference.

Recall that the preference matrix of a recommendation system normally contains multiple nonzero entries in a given user-row; items recommendations are made according to the nonzero entries in the user-row by assuming that the user is 'typical' Drineas et al. [9]. However, in a KG there might be only one nonzero entry in the row (s, p, \cdot) . Therefore, we suggest, for the inference on a KG quantum algorithm needs to sample triples with the given subject s and post-select on the predicate p . Post-selection can be a feasible step if the number of semantic triples with s as subject and p and predicate is $\mathcal{O}(1)$.

Before sketching the algorithm, we need to mention the quantum data structure since our method contains the preparing and exponentiating of a density matrix derived from the tensorized classical data. The most difficult technical challenges of quantum machine learning are loading classical data as quantum states and measuring the states since reading or writing high-dimensional data from quantum states might obliterate the quantum acceleration. Therefore, the technique quantum Random Access Memory (qRAM) Giovannetti et al. [13] was developed, which can load classical data into quantum states with exponential acceleration. Appendix A.3 gives more details on loading vector and tensorized classical data.

The basic idea of our quantum algorithm is to project the observed data onto the eigenspaces of $\hat{\chi}$ whose corresponding singular values are larger than a threshold. Therefore, we need to create an operator that can reveal the eigenspaces and singular values of $\hat{\chi}$. The first step is to prepare the following density matrix from $\hat{\chi}$ via a tensor contraction scheme:

$$\rho_{\hat{\chi}^\dagger \hat{\chi}} := \sum_{i_2 i_3 i'_2 i'_3} \sum_{i_1} \hat{\chi}_{i_1, i_2 i_3}^\dagger \hat{\chi}_{i_1, i'_2 i'_3} |i_2 i_3\rangle \langle i'_2 i'_3|, \quad (4)$$

where $\sum_{i_1} \hat{\chi}_{i_1, i_2 i_3}^\dagger \hat{\chi}_{i_1, i'_2 i'_3}$ means tensor contraction along the first dimension; a normalization factor is neglected temporarily. Later we will elaborate why we perform contraction along the first dimension. We have the following lemma about $\rho_{\hat{\chi}^\dagger \hat{\chi}}$ preparation.

LEMMA 2. $\rho_{\hat{\chi}^\dagger \hat{\chi}}$ can be prepared via qRAM in time $O(\text{polylog}(d_1 d_2 d_3))$.

PROOF. Since $\hat{\chi} \in \mathbb{R}^{d_1 \times d_2 \times d_3}$ is a real-valued tensor, the quantum state $\sum_{i_1 i_2 i_3} \hat{\chi}_{i_1 i_2 i_3} |i_1 i_2 i_3\rangle = \sum_{i_1 i_2 i_3} \hat{\chi}_{i_1 i_2 i_3} |i_1\rangle \otimes |i_2\rangle \otimes |i_3\rangle$ can be prepared via qRAM in time $O(\text{polylog}(d_1 d_2 d_3))$, where $|i_1\rangle \otimes |i_2\rangle \otimes |i_3\rangle$ represents the tensor product of index registers in the canonical basis. The corresponding density matrix of the quantum state reads

$$\rho = \sum_{i_1 i_2 i_3} \sum_{i'_1 i'_2 i'_3} \hat{\chi}_{i_1 i_2 i_3} |i_1\rangle \otimes |i_2\rangle \otimes |i_3\rangle \langle i'_1| \otimes \langle i'_2| \otimes \langle i'_3| \hat{\chi}_{i'_1 i'_2 i'_3}^\dagger.$$

After preparation, a partial trace implemented on the first index register of the density matrix

$$\begin{aligned} \text{tr}_1(\rho) &= \sum_{i_2 i_3} \sum_{i'_2 i'_3} \sum_{i_1} \hat{\chi}_{i_1 i_2 i_3} |i_2\rangle \otimes |i_3\rangle \langle i'_2| \otimes \langle i'_3| \hat{\chi}_{i_1 i'_2 i'_3}^\dagger \\ &= \sum_{i_2 i_3} \sum_{i'_2 i'_3} \hat{\chi}_{i_1 i_2 i_3}^\dagger \hat{\chi}_{i_1 i'_2 i'_3} |i_2 i_3\rangle \langle i'_2 i'_3| \end{aligned}$$

gives the desired operator $\rho_{\hat{\chi}^\dagger \hat{\chi}}$. ■

Suppose that $\hat{\chi}$ has a tensor SVD approximation with $\hat{\chi} \approx \sum_{i=1}^R \sigma_i u_1^{(i)} \otimes u_2^{(i)} \otimes u_3^{(i)}$. Then the spectral decomposition of the density operator can be written as

$$\rho_{\hat{\chi}^\dagger \hat{\chi}} = \frac{1}{\sum_{i=1}^R \sigma_i^2} \sum_{i=1}^R \sigma_i^2 |u_2^{(i)}\rangle \otimes |u_3^{(i)}\rangle \langle u_2^{(i)}| \otimes \langle u_3^{(i)}|.$$

Especially, the eigenstates $|u_2^{(i)}\rangle \otimes |u_3^{(i)}\rangle$ of $\rho_{\hat{\chi}^\dagger \hat{\chi}}$ form another set of basis in the Hilbert space of the tensor product of quantum index registers.

The next step is to readout singular values of $\rho_{\hat{\chi}^\dagger \hat{\chi}}$ and write into another quantum register via the density matrix exponentiation method proposed in Lloyd et al. [22]. This step is also referred to as quantum principal component analysis (qPCA). The key is to prepare the unitary operator

$$U := \sum_{k=0}^{K-1} |k \Delta t\rangle \langle k \Delta t|_C \otimes \exp(-ik \Delta t \tilde{\rho}_{\hat{\chi}^\dagger \hat{\chi}})$$

which is the tensor product of a maximally mixed state $\sum_{k=0}^{K-1} |k \Delta t\rangle \langle k \Delta t|_C$ with the exponentiation of the rescaled density matrix $\tilde{\rho}_{\hat{\chi}^\dagger \hat{\chi}}$. Especially, the clock register C is needed for the phase estimation and Δt determines the precision of estimated singular values. The following Lemma shows that the Hamiltonian simulation with unitary operator $e^{-it \tilde{\rho}_{\hat{\chi}^\dagger \hat{\chi}}}$ can be applied on arbitrary quantum states for any simulation time t .

LEMMA 3. Unitary operator $e^{-it \tilde{\rho}_{\hat{\chi}^\dagger \hat{\chi}}}$ can be applied to any quantum state, where $\tilde{\rho}_{\hat{\chi}^\dagger \hat{\chi}} := \frac{\rho_{\hat{\chi}^\dagger \hat{\chi}}}{d_2 d_3}$, up to simulation time t . The total number of steps for simulation is $O(\frac{t^2}{\epsilon} T_\rho)$, where ϵ is the desired accuracy, and T_ρ is the time for accessing the density matrix.

PROOF. The proof uses the dense matrix exponentiation method proposed in Rebentrost et al. [29], which was developed from Lloyd [21]. One crucial step is to show that Hamiltonian simulation in infinitesimal time step can be

implemented with a simple unitary swap operator without exponentiating the Hamiltonian. Details are in Appendix A.4 and Lemma A 8, 9. ■

The algorithm samples triples with subject s given the query $(s, p, ?)$. Hence a quantum state $|\hat{\chi}_s^{(1)}\rangle_I$ needs to be created first via qRAM in the input data register I , where $\hat{\chi}_s^{(1)}$ denotes the s -row of the flattened tensor $\hat{\chi}$ along the first dimension. After that, the operator U is applied to the quantum state $\sum_{k=0}^{K-1} |k\Delta t\rangle_C \otimes |\hat{\chi}_s^{(1)}\rangle_I$. After this stage of computation, we obtain

$$\sum_{i=1}^R \beta_i \left(\sum_{k=0}^{K-1} e^{-ik\Delta t} \tilde{\sigma}_i^2 |k\Delta t\rangle_C \right) |u_i^{(2)}\rangle_I \otimes |u_i^{(3)}\rangle_I, \quad (5)$$

where $\tilde{\sigma}_i := \frac{\sigma_i}{\sqrt{d_2 d_3}}$ are the rescaled singular values of $\tilde{\rho}_{\hat{\chi}^\dagger \hat{\chi}}$ (see Eq. 3). Moreover, β_i are the coefficients of $|\hat{\chi}_s^{(1)}\rangle_I$ decomposed in the eigenbasis $|u_2^{(i)}\rangle_I \otimes |u_3^{(i)}\rangle_I$ of $\rho_{\hat{\chi}^\dagger \hat{\chi}}$, namely $|\hat{\chi}_s^{(1)}\rangle_I = \sum_{i=1}^R \beta_i |u_2^{(i)}\rangle_I \otimes |u_3^{(i)}\rangle_I$.

The third step is to perform the quantum phase estimation on the clock register C , which is restated in the next Theorem.

THEOREM 3 (QUANTUM PHASE ESTIMATION KITAEV [19]). *Let unitary $U |v_j\rangle = e^{i\theta_j} |v_j\rangle$ with $\theta_j \in [-\pi, \pi]$ for $j \in [n]$. There is a quantum algorithm that transforms $\sum_{j \in [n]} \alpha_j |v_j\rangle \mapsto \sum_{j \in [n]} \alpha_j |v_j\rangle |\bar{\theta}_j\rangle$ such that $|\bar{\theta}_j - \theta_j| \leq \epsilon$ for all $j \in [n]$ with probability $1 - 1/\text{poly}(n)$ in time $O(T_U \log(n)/\epsilon)$, where T_U is the time to implement U .*

The resulting state after phase estimation reads $\sum_{i=1}^R \beta_i |\lambda_i\rangle_C \otimes |u_2^{(i)}\rangle_I \otimes |u_3^{(i)}\rangle_I$ where $\lambda_i := \frac{2\pi}{\tilde{\sigma}_i^2}$. In fact, it can be shown that the probability amplitude of measuring the register C is maximized when $k\Delta t = \lfloor \frac{2\pi}{\tilde{\sigma}_i^2} \rfloor$, where $\lfloor \cdot \rfloor$ represents the nearest integer. Therefore, the small time step Δt determines the accuracy of quantum phase estimation. We chose $\Delta t = O(\frac{1}{\epsilon})$, and according to Lemma 3 the total run time is $O(\frac{1}{\epsilon^3} T_{\tilde{\rho}}) = O(\frac{1}{\epsilon^3} \text{polylog}(d_1 d_2 d_3))$. We also perform controlled computation on the clock register to recover the original singular values of $\rho_{\hat{\chi}^\dagger \hat{\chi}}$, and obtain $\sum_{i=1}^R \beta_i |\sigma_i^2\rangle_C \otimes |u_2^{(i)}\rangle_I \otimes |u_3^{(i)}\rangle_I$.

The next step is to perform quantum singular value projection on the quantum state obtained from the last step. Notice that, classically, this step corresponds to projecting $\hat{\chi}$ onto the subspace $\hat{\chi}_{|\cdot| \geq \tau}$. In this way, observed entries will be smoothed and unobserved entries get boosted from which we can infer unobserved triples $(s, p, ?)$ in the test dataset (see Theorem 2). Quantum singular value projection given the threshold $\tau > 0$ can be implemented in the following way. We first create a new register R using an auxiliary qubit and a unitary operation that maps $|\sigma_i^2\rangle_C \otimes |0\rangle_R$ to $|\sigma_i^2\rangle_C \otimes |1\rangle_R$ only if $\sigma_i^2 < \tau^2$, otherwise $|0\rangle_R$ remains unchanged. This step of projection gives the state

$$\sum_{i: \sigma_i^2 \geq \tau^2} \beta_i |\sigma_i^2\rangle_C \otimes |u_2^{(i)}\rangle_I \otimes |u_3^{(i)}\rangle_I \otimes |0\rangle_R + \sum_{i: \sigma_i^2 < \tau^2} \beta_i |\sigma_i^2\rangle_C \otimes |u_2^{(i)}\rangle_I \otimes |u_3^{(i)}\rangle_I \otimes |1\rangle_R. \quad (6)$$

The last step is to erase the clock register using reversible unitary operator U again; measure the new register R and post-select on the state $|0\rangle_R$; and trace-out the clock register C . This leads the projected state $\sum_{i: \sigma_i^2 \geq \tau^2} \beta_i |u_2^{(i)}\rangle_I \otimes |u_3^{(i)}\rangle_I$.

In summary, implementing all aforementioned quantum operations, in fact, produces $|\hat{\chi}_{|\cdot| \geq \tau}^+ \hat{\chi}_{|\cdot| \geq \tau}^{(1)}\rangle_I$ from the input data state $|\hat{\chi}_s^{(1)}\rangle_I$, where

$$\hat{\chi}_{|\cdot| \geq \tau}^+ \hat{\chi}_{|\cdot| \geq \tau} = \sum_{i: |\sigma_i| \geq \tau} \left(\frac{1}{\sigma_i} u_2^{(i)} \otimes u_3^{(i)} \right) \otimes (\sigma_i u_2^{(i)} \otimes u_3^{(i)})^\top,$$

Algorithm 1 Quantum Tensor SVD on Knowledge Graph**Input:** Inference task (s, p, ?)**Output:** Possible objects to the inference task**Require:** Quantum access to $\hat{\chi}$ stored in a classical memory structure; threshold τ for the singular value projection

- 1: Create $\rho_{\hat{\chi}^\dagger \hat{\chi}}$ via qRAM
- 2: Create state $|\hat{\chi}_s^{(1)}\rangle_I$ on the input data register I via qRAM
- 3: Prepare unitary operator U and apply on $|\hat{\chi}_s^{(1)}\rangle_I$, where

$$U := \sum_{k=0}^{K-1} |k \Delta t\rangle \langle k \Delta t|_C \exp(-ik \Delta t \tilde{\rho}_{\hat{\chi}^\dagger \hat{\chi}})$$

- 4: Quantum phase estimation on the clock register C to obtain $\sum_{i=1}^R \beta_i |\lambda_i\rangle_C \otimes |u_2^{(i)}\rangle_I \otimes |u_3^{(i)}\rangle_I$
- 5: Controlled computation on the clock register C to obtain $\sum_{i=1}^R \beta_i |\sigma_i^2\rangle_C \otimes |u_2^{(i)}\rangle_I \otimes |u_3^{(i)}\rangle_I$
- 6: Singular value projection given the threshold τ to obtain $\sum_{i: \sigma_i^2 \geq \tau^2} \beta_i |\sigma_i^2\rangle_C \otimes |u_2^{(i)}\rangle_I \otimes |u_3^{(i)}\rangle_I \otimes |0\rangle_R + \sum_{i: \sigma_i^2 < \tau^2} \beta_i |\sigma_i^2\rangle_C \otimes |u_2^{(i)}\rangle_I \otimes |u_3^{(i)}\rangle_I \otimes |1\rangle_R$
- 7: Measure on the register R and post-select the state $|0\rangle_R$
- 8: Partial trace over the clock register C
- 9: Measure the resulting state $\sum_{i: |\sigma_i| \geq \tau} \beta_i |u_2^{(i)}\rangle_I \otimes |u_3^{(i)}\rangle_I$ in the canonical basis of the input register I
- 10: Post-select on the predicate p from the sampled triples (s, \cdot , \cdot)

and \cdot^+ represents pseudo-inverse. Now we can recover the ignored normalization factor in Eq. 6 and derive the probability of a successful singular value projection, which is $\frac{\|\hat{\chi}_{|\cdot| \geq \tau} \hat{\chi}_{|\cdot| \geq \tau}^{(1)}\|_2}{\|\hat{\chi}_s^{(1)}\|_2}$. Finally, we measure this state in the canonical basis to get the triples with subject s and post-select on the predicate p. This will return objects to the inference (s, p, ?) after $O(\frac{1}{\epsilon^3} \text{polylog}(d_1 d_2 d_3))$ times of repetitions. The quantum algorithm is summarized in Algorithm 1.

4 EXPERIMENTS WITH CLASSICAL TENSOR SVD

Methods	KINSHIP			FB15k-237		
	MR	@3	@10	MR	@3	@10
RESCAL	3.2	88.8	95.5	291.3	20.7	35.1
TUCKER	2.9	89.8	95.0	276.1	20.9	35.7
COMPLEX	2.2	90.0	97.7	242.7	25.2	39.7
TENSOR SVD	2.7	84.8	96.6	365.5	19.4	35.8

Table 1. Mean Rank, Hits@3, Hits@10 scores of various models compared on the KINSHIP and FB15k-237 datasets.

At the present stage, universal quantum computers are limited by the coherence times of qubits and the fidelity for two-qubit gates. Hence, we investigate the performance of classical tensor SVD on benchmark datasets: KINSHIP and FB15k-237 Toutanova and Chen [33] as the verification of proposed quantum algorithm since it is essentially the quantum counterpart of classical tensor singular value decomposition method. On the other hand, the experiments can additionally verify the primary assumption that the tensor representation of a knowledge graph has a low-rank approximation if the knowledge graph contains global patterns.

Given a semantic triple (s, p, o) , the value function of the tensor SVD is defined as $\eta_{spo} = \sum_{i=1}^R \sigma_i \mathbf{u}_{si} \mathbf{u}_{pi} \mathbf{u}_{oi}$, where $\mathbf{u}_s, \mathbf{u}_p, \mathbf{u}_o$ are R -dimensional vector representations of the subject s , predicate p , and object o , respectively. Intuitively, the value function indicates the class of a given semantic tensor in a binary classification with 1 representing genuine triple, while 0 false triple. Note that the vector representations are read out from separate embedding matrices of subjects, predicates, and objects, and the dimension R serves as a hyperparameter.

The model is optimized by minimizing the following objective function

$$\mathcal{L} := \frac{1}{|\mathcal{D}_{\text{train}}|} \sum_{(s,p,o) \in \mathcal{D}_{\text{train}}} (y_{spo} - \eta_{spo})^2 + \gamma (\|U_s^T U_s - \mathbb{I}_R\|_F + \|U_p^T U_p - \mathbb{I}_R\|_F + \|U_o^T U_o - \mathbb{I}_R\|_F)$$

via stochastic gradient descent, which contains a mean square error loss and a penalization. The hyper-parameter γ is used to encourage the orthonormality of embedding matrices for subjects, predicates, and objects as required by the definition of tensor SVD. We compare the performance of tensor SVD model with other benchmark models, e.g., RESCAL Nickel et al. [24], Tucker, and ComplEx Trouillon et al. [34] in Table 1. In Fig. 1 we plot the training curves of the tensor SVD on FB1k-237 using evaluation metrics Mean Rank and Hits@10¹. It shows that the tensor SVD performs reasonably well for small rank, indicating a plausible assumption on the low-rank approximation of the complete knowledge graph tensor. Hence we can estimate the projection threshold τ according to the Theorem 2.

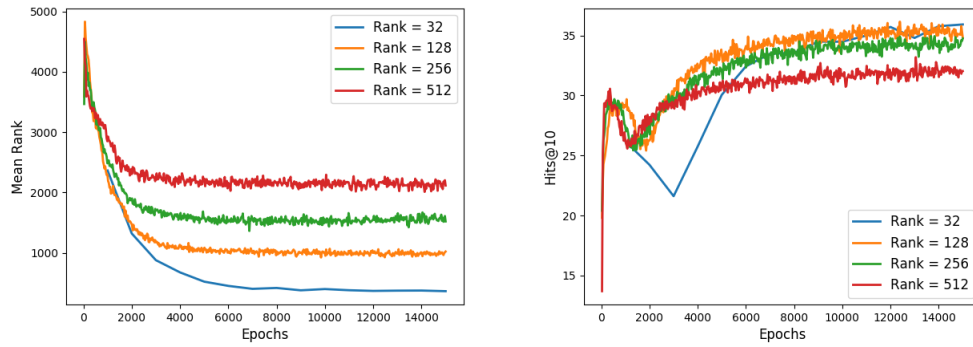


Fig. 1. Mean Rank (left) and Hits@10 (right) scores versus epochs on the FB1k-237 data for different ranks.

5 CONCLUSION

In this work, we presented a quantum machine learning algorithm showing exponentially accelerated inference on knowledge graphs. We first proved that the semantic tensor could approximately be reconstructed from the truncated or projected tensor SVD of the subsampled tensor. Afterward, we constructed a quantum algorithm using quantum principal component analysis and singular value projection. The resulting sample-based quantum machine learning algorithm shows an exponential acceleration w.r.t. the dimensions of the semantic tensor. Due to technical limitations, we study the performance of tensor SVD on classical resources. It shows comparable results to other benchmarking algorithms, which ensures the performance of implementing the quantum tensor SVD on future quantum computers.

¹Details of these evaluation metrics can be found in Bordes et al. [4]. For MR lower is better, while for Hits@10 higher is better.

A APPENDIX

A.1 Proof of Lemma 1

LEMMA A 1 (LEMMA 1 IN THE MAIN TEXT). *If an algorithm returns an approximation of the binary semantic tensor χ , denoted $\hat{\chi}$, with $\|\chi - \hat{\chi}\|_F \leq \epsilon \|\chi\|_F$ and $\epsilon < \frac{1}{2}$, then the probability of a successful information retrieval from the top- n returns of $\hat{\chi}$ is at least $1 - (\frac{\epsilon}{1-\epsilon})^n$.*

PROOF. Since the reconstruction error of χ from $\hat{\chi}$ is upper bounded, we have the following inequality

$$(1 - \epsilon)\|\chi\|_F \leq \|\hat{\chi}\|_F \leq (1 + \epsilon)\|\chi\|_F.$$

We can use this inequality of Frobenius norm to estimate the number of triples which are in $\hat{\chi}$ but not in χ

$$\begin{aligned} \epsilon^2 \|\chi\|_F^2 &\geq \|\chi - \hat{\chi}\|_F^2 = \sum_{(i,j,k) \in \chi \cap (i,j,k) \in \hat{\chi}} (1 - \hat{\chi}_{ijk})^2 + \sum_{(i,j,k) \in \hat{\chi} \cap (i,j,k) \notin \chi} \hat{\chi}_{ijk}^2 + \sum_{(i,j,k) \in \chi \cap (i,j,k) \notin \hat{\chi}} (1 - \hat{\chi}_{ijk})^2 \\ &\geq \sum_{(i,j,k) \in \hat{\chi} \cap (i,j,k) \notin \chi} \hat{\chi}_{ijk}^2, \end{aligned}$$

where we use the notation $(i, j, k) \in \hat{\chi} \cap (i, j, k) \notin \chi$ to represent a semantic triple that can be observed in $\hat{\chi}$ but not in χ , etc. Hence the probability of sampling a semantic triple from $\hat{\chi}$ that doesn't exist in the original tensor is upper bounded by

$$\Pr[(i, j, k) \in \hat{\chi} \cap (i, j, k) \notin \chi] = \frac{\sqrt{\sum_{(i,j,k) \in \hat{\chi} \cap (i,j,k) \notin \chi} \hat{\chi}_{ijk}^2}}{\|\hat{\chi}\|_F} \leq \frac{\epsilon \|\chi\|_F}{\|\hat{\chi}\|_F} \leq \frac{\epsilon}{1 - \epsilon}.$$

Without loss of generality, consider the retrieval of objects given the inference task $(s, p, ?)$. The retrieval becomes unsuccessful if the top- n returns from $\hat{\chi}$ do not contain the correct objects regarding to the query, which has probability at most $(\frac{\epsilon}{1-\epsilon})^n$. Hence the probability of a successful information retrieval from $\hat{\chi}$ is at least $1 - (\frac{\epsilon}{1-\epsilon})^n$. ■

A.2 Proof of Theorem 1 and Theorem 2

We first introduce and recap notations. Consider a N -way tensor $\mathcal{A} \in \mathbb{R}^{d_1 \times d_2 \times \dots \times d_N}$, which has a tensor singular value decomposition with rank R . Let $\mathcal{A}_r = \mathcal{D} \otimes_1 U_1 \otimes_2 U_2 \dots \otimes_N U_N$ denote the truncated r -rank tensor SVD of \mathcal{A} with $U_i = [u_i^{(1)}, \dots, u_i^{(r)}] \in \mathbb{R}^{d_i \times r}$ for $i = 1, \dots, N$ and $\mathcal{D} = \text{diag}(\sigma_1, \dots, \sigma_r) \in \mathbb{R}^{r \times \dots \times r}$. Define the projection operators $\mathcal{P}_i^{\mathcal{A}, r} := \mathbb{I} \otimes \dots \otimes U_i U_i^T \otimes \dots \otimes \mathbb{I}$ with $i = 1, \dots, N$ and the product projections $\mathcal{P}^{\mathcal{A}, r} := \prod_{i=1}^N \mathcal{P}_i^{\mathcal{A}, r}$. We have the following Lemma for the projection operator.

LEMMA A 2. *Consider a tensor \mathcal{A} , if \mathcal{A} has an exact tensor SVD with rank R then $\mathcal{P}^{\mathcal{A}, r} \mathcal{A} = \mathcal{A}_r$. If the tensor SVD of \mathcal{A} is obtained by minimizing $\|\mathcal{A} - \sum_{i=1}^R \sigma_i u_1^{(i)} \otimes u_2^{(i)} \otimes \dots \otimes u_N^{(i)}\|_F := \|\mathcal{A} - \mathcal{A}_R\|_F$, s.t. $\langle u_k^{(i)}, u_k^{(j)} \rangle = \delta_{ij}$ for $k = 1, \dots, N$ with predefined rank R , then $\mathcal{P}^{\mathcal{A}, r} \mathcal{A} = \mathcal{A}_r$ still holds.*

PROOF. We first consider \mathcal{A} has an exact tensor SVD. It means that $\mathcal{A} = \tilde{\mathcal{D}} \otimes_1 \tilde{U}_1 \dots \otimes_N \tilde{U}_N$, where $\tilde{\mathcal{D}} = \text{diag}(\sigma_1, \dots, \sigma_R)$ and $\tilde{U}_i = [u_i^{(1)}, u_i^{(2)}, \dots, u_i^{(R)}]$ for $i = 1, \dots, N$. Hence

$$\mathcal{P}^{\mathcal{A}, r} \mathcal{A} = \tilde{\mathcal{D}} \otimes_1 U_1 U_1^T \tilde{U}_1 \dots \otimes_N U_N U_N^T \tilde{U}_N = \sum_{i=1}^r \sigma_i u_1^{(i)} \otimes_1 u_2^{(i)} \dots \otimes_N u_N^{(i)} = \mathcal{A}_r.$$

On the other hand, suppose that \mathcal{A} 's tensor SVD is found by minimizing the objective function. Define $\mathcal{A}_R^\perp := \mathcal{A} - \mathcal{A}_R$, then we have $\langle \mathcal{A}_R^\perp, \mathcal{T}_i \rangle = 0$ with $\mathcal{T}_i := u_1^{(i)} \otimes u_2^{(i)} \otimes \dots \otimes u_N^{(i)}$ for $i = 1, \dots, R$. To see this, suppose $\exists j$, such that

$\langle \mathcal{A}_R^\perp, \mathcal{T}_j \rangle = \epsilon \neq 0$. Then,

$$\|\mathcal{A} - \sum_{i=1}^R \sigma_i \mathcal{T}_i - \epsilon \mathcal{T}_j\|_F^2 = \|\mathcal{A} - \sum_{i=1}^R \sigma_i \mathcal{T}_i\|_F^2 - \epsilon^2 < \|\mathcal{A} - \sum_{i=1}^R \sigma_i \mathcal{T}_i\|_F^2,$$

which contradicts the fact that \mathcal{A}_R is the global minimum of the objective function. Thus, $\mathcal{P}^{\mathcal{A},r} \mathcal{A} = \mathcal{P}^{\mathcal{A},r}(\mathcal{A}_R + \mathcal{A}_R^\perp) = \mathcal{P}^{\mathcal{A},r} \mathcal{A}_R = \mathcal{A}_r$. ■

As we can see the projection operator $\mathcal{P}^{\mathcal{A},r}$ projects the tensor onto the space spanned by $\mathcal{T}_1, \dots, \mathcal{T}_r$. Lemma A 2 also implies that for any two tensors \mathcal{A} and \mathcal{B} we have the inequality

$$\|\mathcal{P}^{\mathcal{A},r} \mathcal{A}\|_F \geq \|\mathcal{P}^{\mathcal{B},r} \mathcal{A}\|_F. \quad (7)$$

In the next Lemma we give the lower bound of $\|\mathcal{P}^{\mathcal{B},r} \mathcal{A}\|_F$. The proof is similar to the matrix case which is given in Achlioptas and McSherry [1].

LEMMA A 3. *Given two tensors \mathcal{A} and \mathcal{B} having tensor SVD with ranks R_A and R_B , respectively. Suppose $r \leq \min\{R_A, R_B\}$, we have*

$$\|\mathcal{P}^{\mathcal{B},r} \mathcal{A}\|_F \geq \|\mathcal{P}^{\mathcal{A},r} \mathcal{A}\|_F - 2\|\mathcal{P}^{\mathcal{A}-\mathcal{B},r}(\mathcal{A} - \mathcal{B})\|_F.$$

PROOF.

$$\begin{aligned} \|\mathcal{P}^{\mathcal{B},r} \mathcal{A}\|_F &= \|\mathcal{P}^{\mathcal{B},r}(\mathcal{B} + (\mathcal{A} - \mathcal{B}))\|_F \geq \|\mathcal{P}^{\mathcal{B},r} \mathcal{B}\|_F - \|\mathcal{P}^{\mathcal{B},r}(\mathcal{A} - \mathcal{B})\|_F \\ &\geq \|\mathcal{P}^{\mathcal{A},r} \mathcal{B}\|_F - \|\mathcal{P}^{\mathcal{B},r}(\mathcal{A} - \mathcal{B})\|_F = \|\mathcal{P}^{\mathcal{A},r}(\mathcal{A} - (\mathcal{A} - \mathcal{B}))\|_F - \|\mathcal{P}^{\mathcal{B},r}(\mathcal{A} - \mathcal{B})\|_F \\ &\geq \|\mathcal{P}^{\mathcal{A},r} \mathcal{A}\|_F - \|\mathcal{P}^{\mathcal{A},r}(\mathcal{A} - \mathcal{B})\|_F - \|\mathcal{P}^{\mathcal{B},r}(\mathcal{A} - \mathcal{B})\|_F \\ &\geq \|\mathcal{P}^{\mathcal{A},r} \mathcal{A}\|_F - 2\|\mathcal{P}^{\mathcal{A}-\mathcal{B},r}(\mathcal{A} - \mathcal{B})\|_F, \end{aligned}$$

where we used Eq. 7 multiple times. ■

Lemma A 3 indicates that if \mathcal{A} and \mathcal{B} are similar tensors, then the projection of tensor \mathcal{A} onto the first r bases of tensor \mathcal{B} has only small error which is bounded by $\|\mathcal{P}^{\mathcal{A}-\mathcal{B},r}(\mathcal{A} - \mathcal{B})\|_F$. Using Lemma A 3 we can derive the following bound which will serve as the main Lemma for estimating the bound of reconstruction error.

LEMMA A 4. *Given two tensors \mathcal{A} and \mathcal{B} having tensor SVD with ranks R_A and R_B , respectively. Suppose $r \leq \min\{R_A, R_B\}$, we have*

$$\begin{aligned} \|\mathcal{A} - \mathcal{P}^{\mathcal{B},r} \mathcal{B}\|_F &\leq 2\|\mathcal{A} - \mathcal{A}_r\|_F + 2\sqrt{\|\mathcal{A}_r\|_F \|\mathcal{A} - \mathcal{A}_r\|_F} + 2\sqrt{\|\mathcal{A}_r\|_F \|\mathcal{P}^{\mathcal{A}-\mathcal{B},r}(\mathcal{A} - \mathcal{B})\|_F} + \|\mathcal{P}^{\mathcal{A}-\mathcal{B},r}(\mathcal{A} - \mathcal{B})\|_F. \end{aligned}$$

PROOF.

$$\begin{aligned}
\|\mathcal{A} - \mathcal{P}^{\mathcal{B},r}\mathcal{B}\|_F &= \|\mathcal{A} - \mathcal{P}^{\mathcal{B},r}(\mathcal{A} - (\mathcal{A} - \mathcal{B}))\|_F \leq \|\mathcal{A} - \mathcal{P}^{\mathcal{B},r}\mathcal{A}\|_F + \|\mathcal{P}^{\mathcal{B},r}(\mathcal{A} - \mathcal{B})\|_F \\
&\leq \|\mathcal{P}^{\mathcal{A},r}\mathcal{A} - \mathcal{P}^{\mathcal{B},r}\mathcal{A}\|_F + \|\mathcal{A} - \mathcal{P}^{\mathcal{A},r}\mathcal{A}\|_F + \|\mathcal{P}^{\mathcal{B},r}(\mathcal{A} - \mathcal{B})\|_F \\
&= \|\mathcal{A}_r - \mathcal{P}^{\mathcal{B},r}((\mathcal{A} - \mathcal{A}_r) + \mathcal{A}_r)\|_F + \|\mathcal{A} - \mathcal{P}^{\mathcal{A},r}\mathcal{A}\|_F + \|\mathcal{P}^{\mathcal{B},r}(\mathcal{A} - \mathcal{B})\|_F \\
&\leq \|\mathcal{A}_r - \mathcal{P}^{\mathcal{B},r}\mathcal{A}_r\|_F + \|\mathcal{P}^{\mathcal{B},r}(\mathcal{A} - \mathcal{A}_r)\|_F + \|\mathcal{A} - \mathcal{P}^{\mathcal{A},r}\mathcal{A}\|_F + \|\mathcal{P}^{\mathcal{B},r}(\mathcal{A} - \mathcal{B})\|_F \\
&\leq \underbrace{\|\mathcal{A}_r - \mathcal{P}^{\mathcal{B},r}\mathcal{A}_r\|_F}_{(\star)} + 2\|\mathcal{A} - \mathcal{P}^{\mathcal{A},r}\mathcal{A}\|_F + \|\mathcal{P}^{\mathcal{A}-\mathcal{B},r}(\mathcal{A} - \mathcal{B})\|_F,
\end{aligned}$$

for the last inequality we use Eq. 7 multiple times. Now we can apply Pythagorean theorem on the first r eigenbases of tensor \mathcal{A} to bound the term (\star) . Hence

$$\begin{aligned}
(\star) &= \sqrt{\|\mathcal{A}_r\|_F^2 - \|\mathcal{P}^{\mathcal{B},r}\mathcal{A}_r\|_F^2} \\
&\stackrel{(1)}{\leq} \sqrt{\|\mathcal{A}_r\|_F^2 - \|\mathcal{A}_r\|_F^2 + 4\|\mathcal{A}_r\|_F\|\mathcal{P}^{\mathcal{A}-\mathcal{B},r}(\mathcal{A}_r - \mathcal{B})\|_F} \\
&= 2\sqrt{\|\mathcal{A}_r\|_F\|\mathcal{P}^{\mathcal{A}-\mathcal{B},r}(\mathcal{A}_r - \mathcal{B})\|_F} \\
&\leq 2\sqrt{\|\mathcal{A}_r\|_F[\|\mathcal{P}^{\mathcal{A}-\mathcal{B},r}(\mathcal{A}_r - \mathcal{A})\|_F + \|\mathcal{P}^{\mathcal{A}-\mathcal{B},r}(\mathcal{A} - \mathcal{B})\|_F]} \\
&\stackrel{(2)}{\leq} 2\sqrt{\|\mathcal{A}_r\|_F[\|\mathcal{A}_r - \mathcal{A}\|_F + \|\mathcal{P}^{\mathcal{A}-\mathcal{B},r}(\mathcal{A} - \mathcal{B})\|_F]} \\
&\stackrel{(3)}{\leq} 2\sqrt{\|\mathcal{A}_r\|_F\|\mathcal{A} - \mathcal{A}_r\|_F} + 2\sqrt{\|\mathcal{A}_r\|_F\|\mathcal{P}^{\mathcal{A}-\mathcal{B},r}(\mathcal{A} - \mathcal{B})\|_F},
\end{aligned}$$

where inequality (1) is given by Lemma A 3, (2) by Eq. 7 and (3) is according to $\sqrt{x+y} \leq \sqrt{x} + \sqrt{y}$. \blacksquare

In summary, we have the following bound

$$\begin{aligned}
&\|\mathcal{A} - \mathcal{P}^{\mathcal{B},r}\mathcal{B}\|_F \\
&\leq 2\|\mathcal{A} - \mathcal{A}_r\|_F + 2\sqrt{\|\mathcal{A}_r\|_F\|\mathcal{A} - \mathcal{A}_r\|_F} + 2\sqrt{\|\mathcal{A}_r\|_F\|\mathcal{P}^{\mathcal{A}-\mathcal{B},r}(\mathcal{A} - \mathcal{B})\|_F} + \|\mathcal{P}^{\mathcal{A}-\mathcal{B},r}(\mathcal{A} - \mathcal{B})\|_F.
\end{aligned}$$

Consider a tensor \mathcal{A} which will be subsampled and rescaled. The resulting perturbed tensor can be written as $\hat{\mathcal{A}} = \mathcal{A} + \mathcal{N}$, where \mathcal{N} is a noise tensor. In the following, we use $\hat{\mathcal{A}}$ to represent subsampled (sparsified) tensor, and $\hat{\mathcal{A}}_r$ the truncated r -rank tensor SVD of $\hat{\mathcal{A}}$. Thus, according to Lemma A 4 the reconstruction error using the truncated tensor SVD of the sparsified tensor $\hat{\mathcal{A}}$ is upper bounded by

$$\|\mathcal{A} - \hat{\mathcal{A}}_r\|_F \leq 2\|\mathcal{A} - \mathcal{A}_r\|_F + 2\sqrt{\|\mathcal{A}_r\|_F\|\mathcal{A} - \mathcal{A}_r\|_F} + 2\sqrt{\|\mathcal{N}_r\|_F\|\mathcal{A}_r\|_F} + \|\mathcal{N}_r\|_F. \quad (8)$$

To further estimate the bound of the error, we briefly recap the tensor subsampling and sparsification techniques. The basic idea behind matrix/tensor sparsification algorithms is to neglect all small entries, and keep or amplify sufficiently large entries, such that the original matrix/tensor can be reconstructed element-wise with bounded error. Matrix sparsification was first studied in Achlioptas and McSherry [1], and tensor sparsification in Nguyen et al. [23].

Without further specification, we consider the following general sparsification and rescaling method used in the main text:

$$\hat{\mathcal{A}}_{i_1 i_2 \dots i_N} = \begin{cases} \frac{\mathcal{A}_{i_1 i_2 \dots i_N}}{p} & \text{with probability } p > 0 \\ 0 & \text{otherwise,} \end{cases} \quad (9)$$

where the choice of the element-wise sample probability p will be discussed later. Note that the expectation values of the entries of the sparsified tensor read $\mathbb{E}[\hat{\mathcal{A}}_{i_1 i_2 \dots i_N}] = \mathcal{A}_{i_1 i_2 \dots i_N}$. Recall that the perturbation is defined as $\mathcal{N} = \hat{\mathcal{A}} - \mathcal{A}$. Thus, the entries of the noise tensor have zero mean $\mathbb{E}[\mathcal{N}_{i_1 i_2 \dots i_N}] = 0$ and variance $\text{Var}[\mathcal{N}_{i_1 i_2 \dots i_N}] = \mathcal{A}_{i_1 i_2 \dots i_N}^2 (\frac{1}{p} - 1)$.

To bound the norms of the noise tensor \mathcal{N} we also need the following auxiliary lemmas.

LEMMA A 5. Define two functions $f_1(x) = px + \ln(1 - p + p e^{-x})$ and $f_2(x) = px^2/2$. For any $x \in (-\infty, \infty)$ and $0.22 \leq p \leq 1$, we have $f_1(x) \leq f_2(x)$.

PROOF. We first consider the case when $x \geq 0$. First we have $f_1(0) = f_2(0)$ and $f_1'(0) = f_2'(0)$. Since

$$\begin{aligned} 1 - p + p e^{-x} &= (\sqrt{1-p} - \sqrt{e^{-x}})^2 + 2\sqrt{(1-p)e^{-x}} - e^{-x} + p e^{-x} \\ &\geq 2\sqrt{(1-p)e^{-x}} - (1-p)e^{-x}, \end{aligned}$$

we immediately have the following inequality for the second derivatives of $f_1(x)$ and $f_2(x)$,

$$\begin{aligned} f_1''(x) &= \frac{p(1-p)e^{-x}}{(1-p+p e^{-x})^2} \leq \frac{p(1-p)e^{-x}}{(2\sqrt{(1-p)e^{-x}} - (1-p)e^{-x})^2} \\ &\leq \frac{p(1-p)e^{-x}}{(\sqrt{(1-p)e^{-x}})^2} = p = f_2''(x). \end{aligned} \quad (10)$$

We used the condition that $0 \leq p \leq 1$ and $e^{-x} \leq 1$ for $x \geq 0$ to derive the second inequality in Eq. 10. Hence $f_1(x) \leq f_2(x)$ for any $x \geq 0$ and $0 \leq p \leq 1$.

Next, we consider the case when $x < 0$ for different values of p . To find the condition of non-negative p such that $f_1(x) \leq f_2(x)$ we need to solve a transcendent inequality numerically. Hence in Figure 2 we plot $f_1(x) - f_2(x)$ as a function of x and p . From Figure 2 we can read the following numerical conditions

$x = 0.$	$p \geq 0.$	$x = -0.5$	$p \geq 0.1185$	$x = -1$	$p \geq 0.1772$	$x = -6$	$p \geq 0.1787$
$x = -0.1$	$p \geq 0.0310$	$x = -0.6$	$p \geq 0.1337$	$x = -2$	$p \geq 0.2184$	$x = -7$	$p \geq 0.1652$
$x = -0.2$	$p \geq 0.0577$	$x = -0.7$	$p \geq 0.1469$	$x = -3$	$p \geq 0.2196$	$x = -8$	$p \geq 0.1531$
$x = -0.3$	$p \geq 0.0809$	$x = -0.8$	$p \geq 0.1585$	$x = -4$	$p \geq 0.2082$	$x = -10$	$p \geq 0.1331$
$x = -0.4$	$p \geq 0.1010$	$x = -0.9$	$p \geq 0.1685$	$x = -5$	$p \geq 0.1934$	$x = -20$	$p \geq 0.0794$

Table 2. Numerical conditions for non-negative p such that $f_1(x, p) - f_2(x, p) \leq 0$ for different values of x .

Table 2 and Figure 2 indicate that if $p \gtrsim 0.22$ we have $f_1(x) \leq f_2(x)$ for any $x < 0$ in the worst case. In summary, $f_1(x) \leq f_2(x)$ for any $x \in (-\infty, \infty)$ and $0.22 \leq p \leq 1$. ■

LEMMA A 6. Assume that the noise tensor \mathcal{N} is generated by subsampling a binary tensor $\mathcal{A} \in \{0, 1\}^{d_1 \times d_2 \times \dots \times d_N}$ according to Eq. 9 with sample probability $p \gtrsim 0.22$. The spectral norm of \mathcal{N} is bounded by

$$\|\mathcal{N}\|_\sigma \leq \sqrt{\frac{8}{p} \left(\log\left(\frac{2N}{N_0}\right) \sum_{k=1}^N d_k + \log \frac{2}{\delta} \right)}, \quad (11)$$

with probability at least $1 - \delta$.

PROOF. Recall that the noise tensor entries $\mathcal{N}_{i_1 i_2 \dots i_N}$ are independent random variables with zero mean and

$$\mathcal{N}_{i_1 i_2 \dots i_N} = \begin{cases} (\frac{1}{p} - 1)\mathcal{A}_{i_1 i_2 \dots i_N} & \text{with probability } p \\ -\mathcal{A}_{i_1 i_2 \dots i_N} & \text{with probability } 1 - p. \end{cases}$$

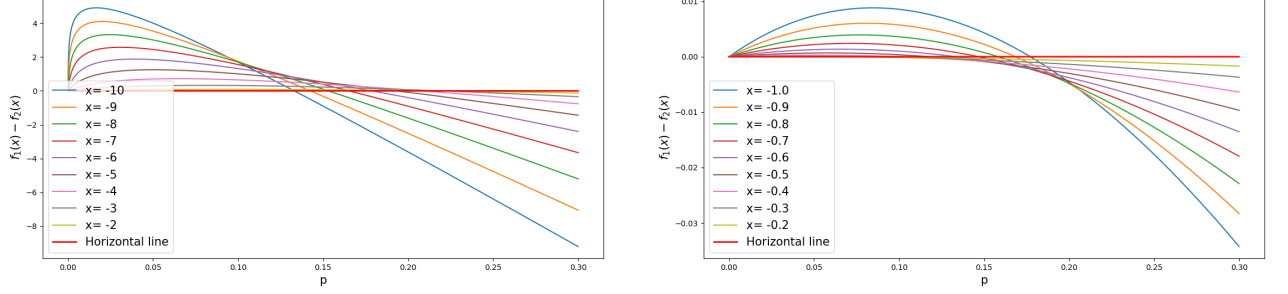


Fig. 2. Plotting $f_1(x, p) - f_2(x, p)$ for $x = [-10, -9, -8, -7, -6, -5, -4, -3, -2]$ (left) and $x = [-1., -0.9, -0.8, -0.7, -0.6, -0.5, -0.4, -0.3, -0.2]$ (right).

We first estimate the quantity $\mathbb{E}[e^{-t\mathcal{N}_{i_1 i_2 \dots i_N} x_{1 i_1} x_{2 i_2} \dots x_{N i_N}}]$ for any $t \geq 0$ with $\mathbf{x}_k \in S^{d_k-1}$, $k = 1, \dots, N$. For the sake of succinct notation we adopt a bijection of index and write $\mathcal{N}_l := \mathcal{N}_{i_1 i_2 \dots i_N}$ and $x_l := x_{1 i_1} x_{2 i_2} \dots x_{N i_N}$ for $l = 1, \dots, d_1 d_2 \dots d_N$. Then we have the following inequality via Lemma A 5

$$\begin{aligned} \mathbb{E}[e^{-t\mathcal{N}_l x_l}] &= p e^{-t(\frac{1}{p}-1)\mathcal{A}_l x_l} + (1-p) e^{t\mathcal{A}_l x_l} = e^{t\mathcal{A}_l x_l} \left(1 - p + p e^{-\frac{t}{p}\mathcal{A}_l x_l}\right) \\ &= e^{py + \ln(1-p+pe^{-y})} \leq e^{\frac{py^2}{2}} \quad \text{for } p \gtrsim 0.22, \end{aligned}$$

where $y := \frac{t\mathcal{A}_l x_l}{p}$. Since $\mathcal{A}_l \in [0, 1]$, we have $\mathbb{E}[e^{-t\mathcal{N}_l x_l}] \leq e^{\frac{t^2}{2p} x_l^2}$ for any $t \geq 0$. In other words, random variables $\mathcal{N}_l x_l$ are sub-Gaussian distributed if the sample probability fulfills $p \gtrsim 0.22$.

Hence

$$\begin{aligned} \mathbb{E}[e^{-t \sum_l \mathcal{N}_l x_l}] &= \mathbb{E}[e^{-t \mathcal{N} \otimes_1 \mathbf{x}_1 \dots \otimes_N \mathbf{x}_N}] \leq \prod_l e^{\frac{t^2}{2p} x_l^2} \\ &= e^{\frac{t^2}{2p} \sum_{i_1=1}^{d_1} x_{1 i_1}^2 \sum_{i_2=1}^{d_2} x_{2 i_2}^2 \dots \sum_{i_N=1}^{d_N} x_{N i_N}^2} = e^{\frac{t^2}{2p}}, \end{aligned}$$

where we use $\|\mathbf{x}_k\|_2 = 1$, $k = 1, \dots, N$.

Given non-negative auxiliary parameters λ and t , we have

$$\begin{aligned} \Pr(\mathcal{N} \otimes_1 \mathbf{x}_1 \dots \otimes_N \mathbf{x}_N \leq -\lambda) &= \Pr(e^{-t \mathcal{N} \otimes_1 \mathbf{x}_1 \dots \otimes_N \mathbf{x}_N} \geq e^{t\lambda}) \\ &\leq e^{-t\lambda} \mathbb{E}[e^{-t \mathcal{N} \otimes_1 \mathbf{x}_1 \dots \otimes_N \mathbf{x}_N}] \\ &\leq e^{\frac{t^2}{2p} - t\lambda} \leq e^{-\frac{p\lambda^2}{2}} \end{aligned}$$

by choosing $t = p\lambda$. Similarly we have the probability $\Pr(\mathcal{N} \otimes_1 \mathbf{x}_1 \dots \otimes_N \mathbf{x}_N \geq \lambda) \leq e^{-\frac{p\lambda^2}{2}}$. In summary,

$$\Pr(|\mathcal{N} \otimes_1 \mathbf{x}_1 \dots \otimes_N \mathbf{x}_N| \geq \lambda) \leq 2e^{-\frac{p\lambda^2}{2}}, \quad (12)$$

if $\mathbf{x}_k \in S^{d_k-1}$, $k = 1, \dots, N$ and $p \geq 0.22$.

Now we are able to use the covering number argument proposed in Tomioka and Suzuki [32] to bound the spectral norm. Let C_1, \dots, C_N be the ϵ -covering of spheres $S^{d_1-1}, \dots, S^{d_N-1}$ with covering number $|C_k|$ upper bounded

by $(\frac{2}{\epsilon})^{d_k}$ for $k = 1, \dots, N$. Since the product space $S^{d_1-1} \times \dots \times S^{d_N-1}$ is closed and bounded, there is a point $(\mathbf{x}_1^*, \dots, \mathbf{x}_N^*) \in S^{d_1-1} \times \dots \times S^{d_N-1}$ which maximizes the tensor-vector product $\mathcal{N} \otimes_1 \mathbf{x}_1 \cdots \otimes_N \mathbf{x}_N$. Hence

$$\|\mathcal{N}\|_\sigma = \mathcal{N} \otimes_1 (\bar{\mathbf{x}}_1 + \boldsymbol{\delta}_1) \cdots \otimes_N (\bar{\mathbf{x}}_N + \boldsymbol{\delta}_N), \quad (13)$$

where $\bar{\mathbf{x}}_k + \boldsymbol{\delta}_k = \mathbf{x}_k^*$ and $\bar{\mathbf{x}}_k \in C_k$ for $k = 1, \dots, N$. According to the definition of ϵ -covering, we have $\|\boldsymbol{\delta}_k\|_2 \leq \epsilon$.

Expanding Eq. 13 gives

$$\|\mathcal{N}\|_\sigma \leq \mathcal{N} \otimes_1 \bar{\mathbf{x}}_1 \cdots \otimes_N \bar{\mathbf{x}}_N + \underbrace{\left(\epsilon N + \epsilon^2 \binom{N}{2} + \dots + \epsilon^N \binom{N}{N} \right)}_{(\star)} \|\mathcal{N}\|_\sigma.$$

Furthermore, we choose $\epsilon = \frac{\log \frac{3}{2}}{N}$ and estimate the above (\star) term as follows

$$(\star) \leq \epsilon N + \frac{(\epsilon N)^2}{2!} + \dots + \frac{(\epsilon N)^N}{N!} \leq e^{\epsilon N} - 1 = \frac{1}{2}.$$

Hence

$$\|\mathcal{N}\|_\sigma \leq 2 \max_{\bar{\mathbf{x}}_k \in C_k, k=1, \dots, N} \mathcal{N} \otimes_1 \bar{\mathbf{x}}_1 \cdots \otimes_N \bar{\mathbf{x}}_N.$$

Using the property of ϵ -covering and Eq. 12 we can derive the following inequality for any $\lambda \geq 0$

$$\begin{aligned} \Pr(\|\mathcal{N}\|_\sigma \geq \lambda) &\leq \Pr(2 \max_{\bar{\mathbf{x}}_k \in C_k, k=1, \dots, N} \mathcal{N} \otimes_1 \bar{\mathbf{x}}_1 \cdots \otimes_N \bar{\mathbf{x}}_N \geq \lambda) \\ &\leq \sum_{\bar{\mathbf{x}}_k \in C_k, k=1, \dots, N} \leq \left(\frac{2}{\epsilon}\right)^{\sum_{k=1}^N d_k} 2e^{-\frac{p\lambda^2}{8}}. \end{aligned}$$

Setting $\Pr(\|\mathcal{N}\|_\sigma \geq \lambda) = \delta$, the spectral norm of the noise tensor \mathcal{N} can be bounded by

$$\|\mathcal{N}\|_\sigma \leq \sqrt{\frac{8}{p} \left(\log\left(\frac{2N}{N_0}\right) \sum_{k=1}^N d_k + \log \frac{2}{\delta} \right)}, \quad N_0 := \log \frac{3}{2} \quad (14)$$

with probability at least $1 - \delta$ if the sample probability satisfies $p \geq 0.22$. \blacksquare

Using $\|\mathcal{N}_r\|_\sigma = \|\mathcal{N}\|_\sigma$, and $\|\mathcal{N}_r\|_F \leq \sqrt{r} \|\mathcal{N}_r\|_\sigma$ we can estimate the norms of the truncated tensor SVD of the noise tensor.

LEMMA A 7.

$$\begin{aligned} \|\mathcal{N}_r\|_\sigma &\leq \sqrt{\frac{8}{p} \left(\log\left(\frac{2N}{N_0}\right) \sum_{k=1}^N d_k + \log \frac{2}{\delta} \right)} \\ \|\mathcal{N}_r\|_F &\leq \sqrt{r \frac{8}{p} \left(\log\left(\frac{2N}{N_0}\right) \sum_{k=1}^N d_k + \log \frac{2}{\delta} \right)}. \end{aligned}$$

where $N_0 = \log \frac{3}{2}$ and the sample probability should satisfy $p \geq 0.22$.

Now we are able to determine the sample probability, such that the error ratio $\frac{\|\mathcal{A} - \hat{\mathcal{A}}_r\|_F}{\|\mathcal{A}\|_F}$ is bounded.

THEOREM A 1 (THEOREM 1 IN THE MAIN TEXT). Let $\mathcal{A} \in \{0, 1\}^{d_1 \times d_2 \times \dots \times d_N}$. Suppose that \mathcal{A} can be well approximated by its r -rank tensor SVD \mathcal{A}_r . Using the subsampling scheme defined in Eq. 9 with the sample probability

$p \geq \max\{0.22, 8r \left(\log(\frac{2N}{N_0}) \sum_{k=1}^N d_k + \log \frac{2}{\delta} \right) / (\tilde{\epsilon} \|\mathcal{A}\|_F)^2\}$, $N_0 = \log \frac{3}{2}$, then the original tensor \mathcal{A} can be reconstructed from the truncated tensor SVD of the subsampled tensor $\hat{\mathcal{A}}$. The error satisfies $\|\mathcal{A} - \hat{\mathcal{A}}_r\|_F \leq \epsilon \|\mathcal{A}\|_F$ with probability at least $1 - \delta$, where ϵ is a function of $\tilde{\epsilon}$. Especially, $\tilde{\epsilon}$ together with the sample probability controls the norm of the noise tensor.

PROOF. Suppose tensor \mathcal{A} can be well approximated by its r -rank tensor SVD, in a sense that $\|\mathcal{A} - \mathcal{A}_r\| \leq \epsilon_0 \|\mathcal{A}\|_F$ for some small $\epsilon_0 > 0$. According to Lemma A 7 if we want the Frobenius norm of the noise tensor \mathcal{N}_r to be bounded by $\tilde{\epsilon} \|\mathcal{A}\|_F$ with $\tilde{\epsilon} > 0$, then the sample probability should satisfy $p \geq \{0.22, \frac{8r \left(\log(\frac{2N}{N_0}) \sum_{k=1}^N d_k + \log \frac{2}{\delta} \right)}{(\tilde{\epsilon} \|\mathcal{A}\|_F)^2}\}$.

Using Eq. 8 we have

$$\|\mathcal{A} - \hat{\mathcal{A}}_r\|_F \leq 2\epsilon_0 \|\mathcal{A}\|_F + 2\sqrt{\epsilon_0} \|\mathcal{A}\|_F + 2\sqrt{\tilde{\epsilon}} \|\mathcal{A}\|_F + \tilde{\epsilon} \|\mathcal{A}\|_F = \epsilon \|\mathcal{A}\|_F,$$

where $\epsilon := 2(\epsilon_0 + \sqrt{\epsilon_0} + \sqrt{\tilde{\epsilon}}) + \tilde{\epsilon}$. ■

Note that in the case where \mathcal{A} is a two-dimensional matrix, the sample probability derived in Achlioptas and McSherry [1] reads $\mathcal{O}(\frac{d_1 + d_2}{\|\mathcal{A}\|_F^2})$. This corresponds the high-dimensional tensor case.

For the later use in the quantum algorithm, instead of considering low-rank approximation of the subsampled tensor, we study the tensor SVD with projected singular values, denoted as $\hat{\mathcal{A}}_{|\cdot| \geq \tau}$. This notation denotes that subsampled tensor $\hat{\mathcal{A}}$ is projected onto the eigenspaces with absolute singular values larger than a threshold. Later, it will be also referred to as the projected tensor SVD of $\hat{\mathcal{A}}$ with threshold τ . The following theorem discusses the choice of sample probability and threshold τ , such that the error ratio $\frac{\|\mathcal{A} - \hat{\mathcal{A}}_{|\cdot| \geq \tau}\|_F}{\|\mathcal{A}\|_F}$ is bounded.

THEOREM A 2 (THEOREM 2 IN THE MAIN TEXT). Let $\mathcal{A} \in \{0, 1\}^{d_1 \times d_2 \times \dots \times d_N}$. Suppose that \mathcal{A} can be well approximated by its r -rank tensor SVD \mathcal{A}_r . Using the subsampling scheme defined in Eq. 9 with the sample probability $p \geq \max\{0.22, p_1 := \frac{l_1 C_0}{(\tilde{\epsilon} \|\mathcal{A}\|_F)^2}, p_2 := \frac{r C_0}{(\tilde{\epsilon} \|\mathcal{A}\|_F)^2}, p_3 := \frac{\sqrt{2r C_0}}{\epsilon_1 \tilde{\epsilon} \|\mathcal{A}\|_F}\}$, with $C_0 = 8 \left(\log(\frac{2N}{N_0}) \sum_{k=1}^N d_k + \log \frac{2}{\delta} \right)$, $N_0 = \log \frac{3}{2}$, where l_1 denotes the largest index of singular values of tensor $\hat{\mathcal{A}}$ with $\sigma_{l_1} \geq \tau$, and choosing the threshold as $0 < \tau \leq \frac{\sqrt{2r C_0}}{p \tilde{\epsilon}}$, then the original tensor \mathcal{A} can be reconstructed from the projected tensor SVD of $\hat{\mathcal{A}}$. The error satisfies $\|\mathcal{A} - \hat{\mathcal{A}}_{|\cdot| \geq \tau}\|_F \leq \epsilon \|\mathcal{A}\|_F$ with probability at least $1 - \delta$, where ϵ is a function of $\tilde{\epsilon}$ and ϵ_1 . Especially, $\tilde{\epsilon}$ together with p_1 and p_2 determine the norm of noise tensor and ϵ_1 together with p_3 control the value of $\hat{\mathcal{A}}$'s singular values that are located outside the projection boundary.

PROOF. Suppose tensor \mathcal{A} can be well approximated by its r -rank tensor SVD, in a sense that $\|\mathcal{A} - \mathcal{A}_r\| \leq \epsilon_0 \|\mathcal{A}\|_F$ for some small $\epsilon_0 > 0$. Define the threshold as $\tau := \kappa \|\hat{\mathcal{A}}\|_F > 0$ for some $\kappa > 0$. Let l_1 denote the largest index of singular values of tensor $\hat{\mathcal{A}}$ with $\sigma_{l_1} \geq \kappa \|\hat{\mathcal{A}}\|_F$, and let l_2 denote the smallest index of singular values of tensor $\hat{\mathcal{A}}$ with $\sigma_{l_2} \leq -\kappa \|\hat{\mathcal{A}}\|_F$. If the threshold τ is large enough, we only need to consider the case $l_1 \ll l_2$. Moreover, we have the following constrain for l_1 and κ :

$$l_1 \cdot \sigma_{l_1}^2 \leq \|\hat{\mathcal{A}}_{l_1}\|_F^2 \leq \|\hat{\mathcal{A}}\|_F^2 \Rightarrow l_1 \cdot \kappa^2 \leq 1. \quad (15)$$

Suppose that the tensor $\hat{\mathcal{A}}$ can be well approximated by the tensor SVD with rank R which is written as $\hat{\mathcal{A}}_R$. Note that the rank R can be much larger than r . We first bound $\|\mathcal{A} - \hat{\mathcal{A}}_{|\cdot| \geq \tau}\|_F$ as follows

$$\begin{aligned} \|\mathcal{A} - \hat{\mathcal{A}}_{|\cdot| \geq \tau}\|_F &\approx \|\mathcal{A} - \hat{\mathcal{A}}_{[0, l_1] \cup [l_2, R]}\|_F = \|\mathcal{A} - (\hat{\mathcal{A}}_R - \hat{\mathcal{A}}_{l_2} + \hat{\mathcal{A}}_{l_1})\|_F \\ &\leq \|\mathcal{A} - \hat{\mathcal{A}}_{l_1}\|_F + \|\hat{\mathcal{A}}_{l_2} - \hat{\mathcal{A}}_R\|_F = \|\mathcal{A} - \hat{\mathcal{A}}_{l_1}\|_F + \|\mathcal{A} - \mathcal{A} + \hat{\mathcal{A}}_{l_2} - \hat{\mathcal{A}}_R\|_F \\ &\leq \|\mathcal{A} - \hat{\mathcal{A}}_{l_1}\|_F + \|\mathcal{A} - \hat{\mathcal{A}}_R\|_F + \|\mathcal{A} - \hat{\mathcal{A}}_{l_2}\|_F \\ &\leq 3\|\mathcal{A} - \hat{\mathcal{A}}_{l_1}\|_F. \end{aligned}$$

Assume $l_1 \ll l_2$ and we only distinguish two cases: $l_2 \gg l_1 \geq r$ and $l_1 < r \ll l_2$.

Suppose $l_1 \geq r$, we have

$$\begin{aligned} \|\mathcal{A} - \hat{\mathcal{A}}_{|\cdot| \geq \tau}\|_F &\leq 3\|\mathcal{A} - \hat{\mathcal{A}}_{l_1}\|_F \\ &\stackrel{(1)}{\leq} 3(2\|\mathcal{A} - \mathcal{A}_{l_1}\|_F + 2\sqrt{\|\mathcal{A}_{l_1}\|_F \|\mathcal{A} - \mathcal{A}_{l_1}\|_F} + 2\sqrt{\|\mathcal{N}_{l_1}\|_F \|\mathcal{A}_{l_1}\|_F} + \|\mathcal{N}_{l_1}\|_F) \\ &\stackrel{(2)}{\leq} 3(2\|\mathcal{A} - \mathcal{A}_r\|_F + 2\sqrt{\|\mathcal{A}\|_F \|\mathcal{A} - \mathcal{A}_{l_1}\|_F} + 2\sqrt{\|\mathcal{N}_{l_1}\|_F \|\mathcal{A}\|_F} + \|\mathcal{N}_{l_1}\|_F), \end{aligned}$$

where inequality (1) is given by Eq. 8 and (2) uses $\|\mathcal{A}_{l_1}\|_F \leq \|\mathcal{A}\|_F$.

According to Lemma A 7 if we want the Frobenius norm $\|\mathcal{N}_{l_1}\|_F$ to be bounded by $\tilde{\epsilon}\|\mathcal{A}\|_F$ with $\tilde{\epsilon} > 0$, then the sample probability should satisfy $p \geq \max\{0.22, p_1 := \frac{l_1 C_0}{(\tilde{\epsilon}\|\mathcal{A}\|_F)^2}\}$ where the constant is defined as $C_0 := 8 \left(\log(\frac{2N}{N_0}) \sum_{k=1}^N d_k + \log \frac{2}{\delta} \right)$ (see Lemma A 7). Finally, under this sample condition we have $\|\mathcal{A} - \hat{\mathcal{A}}_{|\cdot| \geq \tau}\|_F \leq 3(2\epsilon_0 + 2\sqrt{\epsilon_0} + 2\sqrt{\tilde{\epsilon}} + \tilde{\epsilon})\|\mathcal{A}\|_F$ for $l_1 \geq r$.

Before considering the case $l_1 < r \ll l_2$ we first estimate the Frobenius norm of subsampled tensor. $\|\hat{\mathcal{A}}\|_F^2$ can be written as a sum of random variables $X_l := \mathcal{A}_l^2$ for $l = 1, \dots, d_1 d_2 \dots d_N$ using a bijection of indices, namely $X := \|\hat{\mathcal{A}}\|_F^2 = \sum_l X_l$. Moreover, $\mathbb{E}[X_l] = \frac{1}{p} \mathcal{A}_l^2$ and $\mathbb{E}[X] = \frac{1}{p} \|\mathcal{A}\|_F^2$. According to the Chernoff bound

$$\Pr(|X - \mathbb{E}[X]| \geq \delta \mathbb{E}[X]) \leq 2e^{-\frac{\mathbb{E}[X]\delta^2}{3}} \quad \text{for all } 0 < \delta < 1, \quad (16)$$

we have $\Pr(\|\hat{\mathcal{A}}\|_F^2 \geq \frac{1+\delta}{p} \|\mathcal{A}\|_F^2) \leq 2e^{-\frac{\|\mathcal{A}\|_F^2 \delta^2}{3p}}$ for $\delta \in (0, 1)$. Hence $\|\hat{\mathcal{A}}\|_F \leq \sqrt{\frac{2}{p}} \|\mathcal{A}\|_F$ is satisfied with high probability.

In the following, we study the case $l_1 < r \ll l_2$ and fix the sample probability p temporarily. It gives

$$\begin{aligned} \|\mathcal{A} - \hat{\mathcal{A}}_{|\cdot| \geq \tau}\|_F &\leq 3\|\mathcal{A} - \hat{\mathcal{A}}_{l_1}\|_F \leq 3(\|\mathcal{A} - \hat{\mathcal{A}}_r\|_F + \|\hat{\mathcal{A}}_r - \hat{\mathcal{A}}_{l_1}\|_F) \\ &\stackrel{(1)}{\leq} 3(2\|\mathcal{A} - \mathcal{A}_r\|_F + 2\sqrt{\|\mathcal{A}_r\|_F \|\mathcal{A} - \mathcal{A}_r\|_F} + 2\sqrt{\|\mathcal{N}_r\|_F \|\mathcal{A}_r\|_F} + \|\mathcal{N}_r\|_F + \|\hat{\mathcal{A}}_r - \hat{\mathcal{A}}_{l_1}\|_F) \\ &\stackrel{(2)}{\leq} 3(2\|\mathcal{A} - \mathcal{A}_r\|_F + 2\sqrt{\|\mathcal{A}\|_F \|\mathcal{A} - \mathcal{A}_r\|_F} + 2\sqrt{\|\mathcal{N}_r\|_F \|\mathcal{A}\|_F} + \|\mathcal{N}_r\|_F + \underbrace{\sqrt{\frac{2r}{p}} \kappa \|\mathcal{A}\|_F}_{(\star)}), \end{aligned} \quad (17)$$

where inequality (1) is given by Eq. 8 and (2) uses the following estimation

$$\|\hat{\mathcal{A}}_r - \hat{\mathcal{A}}_{l_1}\|_F \leq \sqrt{r - l_1} \tau \leq \sqrt{r} \tau = \sqrt{r} \kappa \|\hat{\mathcal{A}}\|_F \leq \sqrt{\frac{2r}{p}} \kappa \|\mathcal{A}\|_F.$$

Similarly, if we want the Frobenius norm $\|\mathcal{N}_r\|_F$ to be bounded by $\tilde{\epsilon}\|\mathcal{A}\|_F$ with $\tilde{\epsilon} > 0$, then the sample probability should satisfy $p \geq \max\{0.22, p_1 := \frac{rC_0}{(\tilde{\epsilon}\|\mathcal{A}\|_F)^2}\}$ according to Lemma A 7. In order to choose κ , we fix the sample probability p temporarily and use the constraint Eq. 15. It gives

$$l_1 < r = \frac{p(\tilde{\epsilon}\|\mathcal{A}\|_F)^2}{C_0} \Rightarrow \kappa^2 \leq \frac{C_0}{p(\tilde{\epsilon}\|\mathcal{A}\|_F)^2}. \quad (18)$$

We can further control the sum of singular values that are located outside the projection boundary by requiring $(\star) \leq \epsilon_1\|\mathcal{A}\|_F$ for some small $\epsilon_1 > 0$ in Eq. 17. Plug the above inequality of κ into the (\star) term we obtain another condition for the sample probability

$$\sqrt{\frac{2r}{p}}\kappa \leq \epsilon_1 \Rightarrow p \geq \frac{\sqrt{2rC_0}}{\epsilon_1\tilde{\epsilon}\|\mathcal{A}\|_F} := p_3. \quad (19)$$

Therefore, in the case $l_1 < r \ll l_2$ if $p \geq \max\{0.22, p_2 = \frac{rC_0}{(\tilde{\epsilon}\|\mathcal{A}\|_F)^2}, p_3 = \frac{\sqrt{2rC_0}}{\epsilon_1\tilde{\epsilon}\|\mathcal{A}\|_F}\}$ we have $\|\mathcal{A} - \hat{\mathcal{A}}\|_{\cdot, \geq \tau} \leq \epsilon\|\mathcal{A}\|_F$, where $\epsilon := 3(2\epsilon_0 + 2\sqrt{\epsilon_0} + 2\sqrt{\tilde{\epsilon}} + \tilde{\epsilon} + \epsilon_1)$.

In summary, combine two situations we have $\|\mathcal{A} - \hat{\mathcal{A}}\|_{\cdot, \geq \tau} \leq \epsilon\|\mathcal{A}\|_F$, where $\epsilon := 3(2\epsilon_0 + 2\sqrt{\epsilon_0} + 2\sqrt{\tilde{\epsilon}} + \tilde{\epsilon} + \epsilon_1)$ if the sample probability is chosen as

$$p \geq \max\{0.22, p_1 = \frac{l_1C_0}{(\tilde{\epsilon}\|\mathcal{A}\|_F)^2}, p_2 = \frac{rC_0}{(\tilde{\epsilon}\|\mathcal{A}\|_F)^2}, p_3 = \frac{\sqrt{2rC_0}}{\epsilon_1\tilde{\epsilon}\|\mathcal{A}\|_F}\}.$$

Moreover, the threshold can be determined from the following approximation after choosing the sample probability:

$$\tau = \kappa\|\hat{\mathcal{A}}\|_F \leq \sqrt{\frac{C_0}{p\tilde{\epsilon}^2}} \frac{\|\hat{\mathcal{A}}\|_F}{\|\mathcal{A}\|_F} \leq \frac{\sqrt{2C_0}}{p\tilde{\epsilon}},$$

where the inequality is derived by using Eq. 18 and $\|\hat{\mathcal{A}}\|_F \leq \sqrt{\frac{2}{p}}\|\mathcal{A}\|_F$. ■

The above estimation on the error bound in the case of projected tensor SVD is crucial for the quantum algorithm since quantum singular value projection depends only on the positive threshold defined for the singular values.

A.3 Data Structure

THEOREM A 3. *Prakash [27] Let $\mathbf{x} \in \mathbb{R}^R$ be a real-valued vector. The quantum state $|x\rangle = \frac{1}{\|\mathbf{x}\|_2} \sum_{i=1}^R x_i |i\rangle$ can be prepared using $\lceil \log_2 R \rceil$ qubits in time $O(\log_2 R)$.*

Theorem A 3 claims that there exist a classical memory structure and a quantum algorithm which can load classical data into a quantum state with exponential acceleration. Figure 3 illustrates a simple example. Given an $R = 4$ dimensional real-valued vector, the quantum state $|x\rangle = x_1|00\rangle + x_2|01\rangle + x_3|10\rangle + x_4|11\rangle$ can be prepared by querying the classical memory structure and applying 3 controlled rotations.

Let us assume that \mathbf{x} is normalized, namely $\|\mathbf{x}\|_2 = 1$. The quantum state $|x\rangle$ is created from the initial state $|0\rangle|0\rangle$ by querying the memory structure from the root to the leaf. The first rotation is applied on the first qubit, giving

$$(\cos \theta_1 |0\rangle + \sin \theta_1 |1\rangle) |0\rangle = (\sqrt{x_1^2 + x_2^2} |0\rangle + \sqrt{x_3^2 + x_4^2} |1\rangle) |0\rangle,$$

where $\theta_1 := \tan^{-1} \sqrt{\frac{x_3^2 + x_4^2}{x_1^2 + x_2^2}}$. The second rotation is applied on the second qubit conditioned on the state of qubit 1. It gives

$$\sqrt{x_1^2 + x_2^2} |0\rangle \frac{1}{\sqrt{x_1^2 + x_2^2}} (|x_1| |0\rangle + |x_2| |1\rangle) + \sqrt{x_3^2 + x_4^2} |1\rangle \frac{1}{\sqrt{x_3^2 + x_4^2}} (|x_3| |0\rangle + |x_4| |1\rangle).$$

The last rotation loads the signs of coefficients conditioned on qubits 1 and 2. In general, an R -dimensional real-valued vector needs to be stored in a classical memory structure with $\lceil \log_2 R \rceil + 1$ layers. The data vector can be loaded into a quantum state using $O(\log_2 R)$ non-trivial controlled rotations.

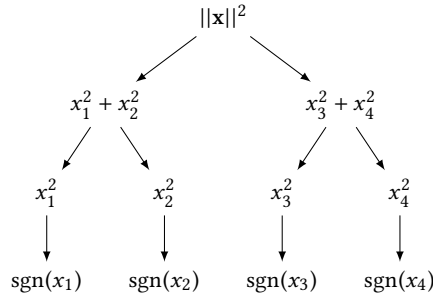


Fig. 3. Classical memory structure with quantum access for creating the quantum state $|x\rangle = x_1 |00\rangle + x_2 |01\rangle + x_3 |10\rangle + x_4 |11\rangle$.

The above simple example of quantum Random Access Memory for generating quantum state from a real-valued vector can be generalized to quantum access of other more complicated data structures, e.g., matrices, tensors.

A.4 Simulation of the unitary operator $e^{-it\tilde{\rho}_{\hat{x}^\dagger \hat{x}}}$

Before proving Lemma A 9 of unitary operator simulation in the main text we give the following auxiliary Lemma. The difficulty of simulating a unitary operator $e^{-i\rho t}$ up to time t is to efficiently exponentiate the density matrix ρ . In Lloyd [21] Lloyd suggested an efficient algorithm for Hamiltonian simulation using a tensor product structure. In particular, the unitary operator $e^{-i\rho\Delta t}$ with small simulation time Δt can be constructed via a simple swap operator.

LEMMA A 8. *Let ρ and σ be density matrices, and S a swap operator such that $S|x\rangle|y\rangle = |y\rangle|x\rangle$. Then for an infinitesimal simulation step Δt we have $e^{-i\rho\Delta t}\sigma e^{i\rho\Delta t} = \text{tr}_1\{e^{-iS\Delta t}\rho \otimes \sigma e^{iS\Delta t}\}$ up to the first order of Δt , where tr_1 is a partial trace applied on the first subsystem of the tensor product structure.*

PROOF. First note that density matrices ρ and σ can be written in the eigenbasis as $\rho = \sum_i |\rho_i\rangle\langle\rho_i|$ and $\sigma = \sum_j |\sigma_j\rangle\langle\sigma_j|$. Moreover, for $\Delta t \rightarrow 0$ we have approximations $e^{-iS\Delta t} \approx \cos \Delta t I - i \sin \Delta t S$ and $e^{iS\Delta t} \approx \cos \Delta t I + i \sin \Delta t S$, where I denotes the identity operator.

Hence

$$\begin{aligned} \text{tr}_1\{e^{-iS\Delta t}\rho \otimes \sigma e^{iS\Delta t}\} &= \text{tr}_1\{(\cos \Delta t I - i \sin \Delta t S)(\sum_{ij} |\rho_i\rangle\langle\rho_j| |\sigma_j\rangle\langle\sigma_i|)(\cos \Delta t I + i \sin \Delta t S)\} \\ &= \text{tr}_1\left\{\sum_{ij} [\cos^2 \Delta t |\rho_i\rangle\langle\rho_j| |\sigma_j\rangle\langle\sigma_i| - i \sin \Delta t \cos \Delta t S |\rho_i\rangle\langle\rho_j| |\sigma_j\rangle\langle\sigma_i| \right. \\ &\quad \left. + i \cos \Delta t \sin \Delta t |\rho_i\rangle\langle\rho_j| |\sigma_j\rangle\langle\sigma_i| S^\dagger + \sin^2 \Delta t S |\rho_i\rangle\langle\rho_j| |\sigma_j\rangle\langle\sigma_i| S^\dagger]\right\} \end{aligned}$$

Recall that $S|\rho_i\rangle|\sigma_j\rangle = |\sigma_j\rangle|\rho_i\rangle$ and $\langle\rho_i|\langle\sigma_j|S^\dagger = \langle\sigma_j|\langle\rho_i|$. Applying the swap operator S gives

$$\text{tr}_1\{e^{-iS\Delta t}\rho \otimes \sigma e^{iS\Delta t}\} \approx \sigma - i\Delta t[\sum_{ij}\langle\rho_i|\sigma_j\rangle|\rho_i\rangle\langle\sigma_j| - \sum_{ij}\langle\sigma_j|\rho_i\rangle|\sigma_j\rangle\langle\rho_i|] + O(\Delta t^2),$$

where we used $\cos \Delta t \approx 1$ and $\sin \Delta t \approx \Delta t$ as $\Delta t \rightarrow 0$. The commutator of two operators is defined as

$$[\rho, \sigma] := \rho\sigma - \sigma\rho = \sum_{ij}\langle\rho_i|\sigma_j\rangle|\rho_i\rangle\langle\sigma_j| - \sum_{ij}\langle\sigma_j|\rho_i\rangle|\sigma_j\rangle\langle\rho_i|$$

and we finally have

$$\text{tr}_1\{e^{-iS\Delta t}\rho \otimes \sigma e^{iS\Delta t}\} = \sigma - i\Delta t[\rho, \sigma] + O(\Delta t^2). \quad (20)$$

On the other hand, applying the limits $\lim_{\Delta t \rightarrow 0} e^{-i\rho\Delta t} = I - i\rho\Delta t$ and $\lim_{\Delta t \rightarrow 0} e^{i\rho\Delta t} = I + i\rho\Delta t$ we can derive

$$\begin{aligned} e^{-i\rho\Delta t}\sigma e^{i\rho\Delta t} &\approx (I - i\Delta t \sum_i |\rho_i\rangle\langle\rho_i|) \sum_j |\sigma_j\rangle\langle\sigma_j| (I + i\Delta t \sum_i |\rho_i\rangle\langle\rho_i|) \\ &= \sigma - i\Delta t[\rho, \sigma] + O(\Delta t^2). \end{aligned}$$

In summary, we have $e^{-i\rho\Delta t}\sigma e^{i\rho\Delta t} = \text{tr}_1\{e^{-iS\Delta t}\rho \otimes \sigma e^{iS\Delta t}\}$ up to the first order of Δt . The above proof indicates that we can construct the unitary operator $e^{-i\rho t}$ and act on the density σ by repeatedly applying simple operations $e^{-iS\Delta t} \approx I - iS\Delta t$ on the tensor product state $\rho \otimes \sigma$ in $n = \frac{t}{\Delta t}$ steps. ■

LEMMA A 9 (LEMMA 3 IN THE MAIN TEXT). *Unitary operator $e^{-it\tilde{\rho}_{\hat{\chi}^\dagger\hat{\chi}}}$ can be applied to any quantum state, where $\tilde{\rho}_{\hat{\chi}^\dagger\hat{\chi}} := \frac{\rho_{\hat{\chi}^\dagger\hat{\chi}}}{d_2d_3}$, up to simulation time t . The total number of steps for simulation is $O(\frac{t^2}{\epsilon}T_\rho)$, where ϵ is the desired accuracy, and T_ρ is the time for accessing the density matrix.*

PROOF. The proof uses the dense matrix exponentiation method proposed in Rebentrost et al. [29] which was developed from Lloyd [21]. Recall that $\rho_{\hat{\chi}^\dagger\hat{\chi}} = \sum_{i_2i_3i'_2i'_3} C_{i_2i_3i'_2i'_3} |i_2i_3\rangle\langle i'_2i'_3|$, where $C_{i_2i_3i'_2i'_3} = \sum_{i_1} \hat{\chi}_{i_1,i_2i_3}^\dagger \hat{\chi}_{i_1,i'_2i'_3}$. For the sake of simplicity, we rewrite $\rho_{\hat{\chi}^\dagger\hat{\chi}}$ as $A \in \mathbb{R}^{N^2 \times N^2}$, where $N := d_2d_3$. Suppose that the unitary operator needs to be applied on the quantum state $|x\rangle$ whose density matrix reads $\sigma := |x\rangle\langle x|$. Then follow the method in Rebentrost et al. [29], we first create a modified swap operator

$$S_A = \sum_{j,k=1}^N A_{jk} |k\rangle\langle j| \otimes |j\rangle\langle k|,$$

and another auxiliary density matrix $\mu = |\vec{1}\rangle\langle\vec{1}|$, with $|\vec{1}\rangle := \frac{1}{\sqrt{N}} \sum_{k=1}^N |k\rangle$. Consider the evolution of the system $\mu \otimes \sigma$ under the unitary operator $e^{-iS_A\Delta t}$ for a small step Δt . With Lemma A 8 it can be shown that

$$\text{tr}_1\{e^{-iS_A\Delta t}\mu \otimes \sigma e^{iS_A\Delta t}\} \approx e^{-i\frac{\Delta}{N}\Delta t} \sigma e^{i\frac{\Delta}{N}\Delta t}.$$

Moreover, repeated applications of $e^{-iS_A\Delta t}$, say n times with $t := n\Delta t$, on the bigger system $\mu \otimes \sigma$ can give $e^{-i\frac{\Delta}{N}t} \sigma e^{i\frac{\Delta}{N}t}$ with is the density matrix of the quantum state $e^{-i\frac{\Delta}{N}t} |x\rangle$. In other words, we can simulate the unitary operator $e^{-it\tilde{\rho}_{\hat{\chi}^\dagger\hat{\chi}}}$ with $\tilde{\rho}_{\hat{\chi}^\dagger\hat{\chi}} := \frac{\rho_{\hat{\chi}^\dagger\hat{\chi}}}{d_2d_3}$.

Furthermore, Rebentrost et al. [29] shows that given t and the required accuracy ϵ , the step size Δt should be small enough, such that $n = O(\frac{t^2}{\epsilon})$. In addition, the quantum access for obtaining the density $\rho_{\hat{\chi}^\dagger\hat{\chi}}$ and creating the modified swap operator requires $T_\rho = O(\text{polylog}(d_1d_2d_3))$ steps. In summary, the total run time for simulating $e^{-it\tilde{\rho}_{\hat{\chi}^\dagger\hat{\chi}}} |x\rangle$ is $nT_\rho = O(\frac{t^2}{\epsilon} \text{polylog}(d_1d_2d_3))$. ■

REFERENCES

- [1] Dimitris Achlioptas and Frank McSherry. Fast computation of low-rank matrix approximations. *Journal of the ACM (JACM)*, 54(2):9, 2007.
- [2] Jacob Biamonte, Peter Wittek, Nicola Pancotti, Patrick Rebentrost, Nathan Wiebe, and Seth Lloyd. Quantum machine learning. *Nature*, 549(7671):195, 2017.
- [3] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250. ACM, 2008.
- [4] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In *Advances in neural information processing systems*, pages 2787–2795, 2013.
- [5] Jie Chen and Yousef Saad. On the tensor svd and the optimal low rank orthogonal approximation of tensors. *SIAM Journal on Matrix Analysis and Applications*, 30(4):1709–1734, 2009.
- [6] Siddhartha Das, George Siopsis, and Christian Weedbrook. Continuous-variable quantum gaussian process regression and quantum singular value decomposition of nonsparse low-rank matrices. *Physical Review A*, 97(2):022315, 2018.
- [7] Lieven De Lathauwer, Bart De Moor, and Joos Vandewalle. A multilinear singular value decomposition. *SIAM journal on Matrix Analysis and Applications*, 21(4):1253–1278, 2000.
- [8] Xin Dong, Evgeniy Gabrilovich, Jeremy Heitz, Wilko Horn, Ni Lao, Kevin Murphy, Thomas Strohmann, Shaohua Sun, and Wei Zhang. Knowledge vault: A web-scale approach to probabilistic knowledge fusion. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 601–610. ACM, 2014.
- [9] Petros Drineas, Iordanis Kerenidis, and Prabhakar Raghavan. Competitive recommendation systems. In *Proceedings of the thirty-fourth annual ACM symposium on Theory of computing*, pages 82–90. ACM, 2002.
- [10] Yuxuan Du, Tongliang Liu, Yinan Li, Runyao Duan, and Dacheng Tao. Quantum divide-and-conquer anchoring for separable non-negative matrix factorization. *IJCAI 2018*, 2018.
- [11] Richard P Feynman. Simulating physics with computers. *International journal of theoretical physics*, 21(6):467–488, 1982.
- [12] Alan Frieze, Ravi Kannan, and Santosh Vempala. Fast monte-carlo algorithms for finding low-rank approximations. *Journal of the ACM (JACM)*, 51(6):1025–1041, 2004.
- [13] Vittorio Giovannetti, Seth Lloyd, and Lorenzo Maccone. Quantum random access memory. *Physical review letters*, 100(16):160501, 2008.
- [14] Lejia Gu, Wang Xiaoqiang, and Zhang Guofeng. Quantum higher order singular value decomposition.
- [15] Aram W Harrow, Avinandan Hassidim, and Seth Lloyd. Quantum algorithm for linear systems of equations. *Physical review letters*, 103(15):150502, 2009.
- [16] Christopher J Hillar and Lek-Heng Lim. Most tensor problems are np-hard. *Journal of the ACM (JACM)*, 60(6):45, 2013.
- [17] Iordanis Kerenidis and Anupam Prakash. Quantum recommendation systems. *arXiv preprint arXiv:1603.08675*, 2016.
- [18] Iordanis Kerenidis, Jonas Landman, Alessandro Luongo, and Anupam Prakash. q-means: A quantum algorithm for unsupervised machine learning. *NeurIPS 2019*, 2019.
- [19] A Yu Kitaev. Quantum measurements and the abelian stabilizer problem. *arXiv preprint quant-ph/9511026*, 1995.
- [20] Tamara G Kolda. Orthogonal tensor decompositions. *SIAM Journal on Matrix Analysis and Applications*, 23(1):243–255, 2001.
- [21] Seth Lloyd. Universal quantum simulators. *Science*, pages 1073–1078, 1996.
- [22] Seth Lloyd, Masoud Mohseni, and Patrick Rebentrost. Quantum principal component analysis. *Nature Physics*, 10(9):631, 2014.
- [23] Nam H Nguyen, Petros Drineas, and Trac D Tran. Tensor sparsification via a bound on the spectral norm of random tensors. *arXiv preprint arXiv:1005.4732*, 2010.
- [24] Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. A three-way model for collective learning on multi-relational data. In *ICML*, volume 11, pages 809–816, 2011.
- [25] Maximilian Nickel, Kevin Murphy, Volker Tresp, and Evgeniy Gabrilovich. A review of relational machine learning for knowledge graphs. *Proceedings of the IEEE*, 104(1):11–33, 2016.
- [26] Maximilian Nickel, Lorenzo Rosasco, Tomaso A Poggio, et al. Holographic embeddings of knowledge graphs. In *AAAI*, volume 2, pages 3–2, 2016.
- [27] Anupam Prakash. *Quantum algorithms for linear algebra and machine learning*. PhD thesis, UC Berkeley, 2014.
- [28] Patrick Rebentrost, Masoud Mohseni, and Seth Lloyd. Quantum support vector machine for big data classification. *Physical review letters*, 113(13):130503, 2014.
- [29] Patrick Rebentrost, Adrian Steffens, Iman Marvian, and Seth Lloyd. Quantum singular-value decomposition of nonsparse low-rank matrices. *Physical review A*, 97(1):012327, 2018.
- [30] Fabian M Suchanek, Gjergji Kasneci, and Gerhard Weikum. Yago: a core of semantic knowledge. In *Proceedings of the 16th international conference on World Wide Web*, pages 697–706. ACM, 2007.
- [31] Ewin Tang. A quantum-inspired classical algorithm for recommendation systems. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, pages 217–228. ACM, 2019.
- [32] Ryota Tomioka and Taiji Suzuki. Spectral norm of random tensors. *arXiv preprint arXiv:1407.1870*, 2014.
- [33] Kristina Toutanova and Danqi Chen. Observed versus latent features for knowledge base and text inference. In *Proceedings of the 3rd Workshop on Continuous Vector Space Models and their Compositionality*, pages 57–66, 2015.

- [34] Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. Complex embeddings for simple link prediction. In *International Conference on Machine Learning*, pages 2071–2080, 2016.
- [35] Nathan Wiebe, Ashish Kapoor, and Krysta Svore. Quantum algorithms for nearest-neighbor methods for supervised and unsupervised learning. *arXiv preprint arXiv:1401.2142*, 2014.
- [36] Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. Embedding entities and relations for learning and inference in knowledge bases. *arXiv preprint arXiv:1412.6575*, 2014.
- [37] Tong Zhang and Gene H Golub. Rank-one approximation to high order tensors. *SIAM Journal on Matrix Analysis and Applications*, 23(2):534–550, 2001.

Chapter 6

Causal Inference under Networked Interference and Intervention Policy Enhancement

Causal Inference under Networked Interference and Intervention Policy Enhancement

Yunpu Ma · Yuyi Wang · Volker Tresp

the date of receipt and acceptance should be inserted later

Abstract Estimating individual treatment effects from data of randomized experiments is a critical task in causal inference. The Stable Unit Treatment Value Assumption (SUTVA) is usually made in causal inference. However, interference can introduce bias when the assigned treatment on one unit affects the potential outcomes of the neighboring units. This interference phenomenon is known as spillover effect in economics or peer effect in social science. Usually, in randomized experiments or observational studies with interconnected units, one can only observe treatment responses under interference. Hence, the issue of how to estimate the superimposed causal effect and recover the individual treatment effect in the presence of interference becomes a challenging task in causal inference. In this work, we study causal effect estimation under general network interference using Graph Neural Networks, which are powerful tools for capturing node and link dependencies in graphs. After deriving causal effect estimators, we further study intervention policy improvement on the graph under capacity constraint. We give policy regret bounds under network interference and treatment capacity constraint. Furthermore, a heuristic graph structure-dependent error bound for Graph Neural Network-based causal estimators is provided.

Keywords Causal Inference · Causal Interference · Graph Neural Network

1 Introduction

Common assumptions made in causal inference are the consistency and interference-free assumptions, i.e., the Stable Unit Treatment Value Assumption (SUTVA) [33], under which the individual treatment response is consistently defined and unaffected by variations in other

Yunpu Ma
Ludwig Maximilian University of Munich
E-mail: cognitive.yunpu@gmail.com

Yuyi Wang
ETH Zurich

Volker Tresp
Ludwig Maximilian University of Munich & Siemens CT
E-mail: volker.tresp@siemens.com

individuals. However, this assumption is problematic under a social network setting since peers are not independent; “no man is an island,” as written by the poet John Donne.

Interference occurs when the treatment response of an individual is influenced through the exposure to its social contacts’ treatments or affected by its social neighbors’ outcomes through peer effects [5, 40]. For instance, the treatment effect of an individual under a vaccination against an infectious disease might influence the health conditions of its surrounding individuals; or a personalized online advertisement might affect other individuals’ purchase of the advertised item through opinion propagation on social networks. Separating individual treatment effect and peer effect in causal inference becomes a difficult problem under interference since, in randomized experiments or observational studies, one can only observe the superposition of both effects. The issue of how to estimate causal responses and make optimal policies on the network is studied in this work.

One of the main objectives of treatment effect estimation is to make better treatment decision rules for individuals according to their characteristics. Population-averaged utility functions have been studied in [27, 3, 19, 20]. In those publications, a policy learner can adapt and improve its decision rules through the utility function. However, interactions among units are always ignored. On the other hand, a policy learner usually faces a capacity or budget constraint, as studied in [22]. Therefore, in this work, we develop a new type of utility function defined on interconnected units and investigate provable policy improvement with budget constraints.

1.1 Related Work

Causal inference with interference was studied in [15, 39, 26]. However, the assumption of group-level interference, having partial interference within the groups and independence across different groups, is often invalid. Hence, several works focus on unit-level causal effects under cross-unit interference and arbitrary treatment assignments, such as [2, 9, 29, 42]. Other approaches for estimating causal effects on networks use graphical models, which are studied in [1, 38].

1.2 Notations and Previous Approaches

Let $\mathcal{G} = (\mathcal{N}, \mathcal{E}, A)$ denote a directed or undirected graph with a node set \mathcal{N} of size n , an edge set \mathcal{E} , and an adjacency matrix $A \in \{0, 1\}^{n \times n}$. For a node, or unit, $i \in \mathcal{N}$, let \mathcal{N}_i indicate the set of neighboring nodes with $A_{ij} = 1$ excluding the node i itself, and let \mathbf{X}_i denote the covariate vector of node i which is defined in the space \mathcal{X} . Let’s first focus on the Neyman–Rubin causal inference model [32, 36]. Let T_i be a binary variable with $T_i = 1$ indicating that node i is in the treatment group, and $T_i = 0$ if i is in the control group. Moreover, let Y_i be the outcome variable with $Y_i(T_i = 1)$ indicating the potential outcome of i under treatment $T_i = 1$ and $Y_i(T_i = 0)$ the potential outcome under control $T_i = 0$. Moreover, we use $T_{\mathcal{N}_i}$ and $Y_{\mathcal{N}_i}$ to represent the treatment assignments and potential outcomes of neighboring nodes \mathcal{N}_i , and \mathbf{T} the entire treatment assignments vector.

In the SUTVA assumption, the individual treatment effect on node i is defined as the difference between outcomes under treatment and under control, i.e., $\tau(\mathbf{X}_i) := \mathbb{E}[Y_i(T_i = 1) - Y_i(T_i = 0) | \mathbf{X}_i]$. To estimate treatment effects under network interference, an exposure variable G is proposed in [40, 5, 2]. The exposure variable G_i is a function of neighboring

treatments $T_{\mathcal{N}_i}$. For instance, G_i can be a variable indicating the level of exposure to the treated neighbors, i.e., $G_i := \frac{\sum_{j \in \mathcal{N}_i} T_j}{|\mathcal{N}_i|}$.

Under the assumption that the outcome only depends on the individual treatment and neighborhood treatments, [9] defines an individual treatment effect under the exposure $G_i = g$ as

$$\tau(\mathbf{X}_i, G_i = g) := \mathbb{E}[Y_i(T_i = 1, G_i = g) - Y_i(T_i = 0, G_i = g) | \mathbf{X}_i]. \quad (1)$$

Moreover, the spillover effect under the treatment $T_i = t$ and the exposure $G_i = g$ is defined as

$$\delta(\mathbf{X}_i, T_i = t, G_i = g) := \mathbb{E}[Y_i(T_i = t, G_i = g) - Y_i(T_i = t, G_i = 0) | \mathbf{X}_i].$$

Treatment and spillover effects are then estimated using generalized propensity score (GPS) weighted estimators.

In general, the outcome model can be more complicated, depending on network topology and covariates of neighboring units. [29] investigates more general causal structural equations under dimension-reducing assumption, and the potential outcome reads $Y_{i,t} := f_Y(\mathbf{X}_i, s_X(\{\mathbf{X}_j | j \in \mathcal{N}_i\}), T_i, s_T(\{T_j | j \in \mathcal{N}_i\}))$, where s_X and s_T are summary functions of neighborhood covariates and treatment, e.g., they could be the summation or average of neighboring treatment assignments and covariates, respectively. Motivated by the above causal structural equation model, we incorporate Graph Neural Network (GNN)-based causal estimators with appropriate covariates and treatment aggregation functions as inputs. GNNs can learn and aggregate feature information from distant neighbors, which makes it a right candidate for capturing the spillover effect given by the neighboring units.

Contributions This work has four major contributions. First, we propose GNN-based causal estimators for causal effect prediction and to recover direct treatment effect under interference (Section 2). Second, we define a novel utility function for policy optimization on a network and derive a graph-dependent policy regret bound (Section 3). Third, we provide an error bound for the GNN-based causal estimators (Section 3 and Appendix F). Last, we conduct extensive experiments to verify the superiority of GNN-based causal estimators and show that the accuracy of a causal estimator is crucial for finding the optimal policy (Section 4).

2 GNN-based Causal Estimators

In this section, we introduce our Graph Neural Network-based causal effect estimators under general network interference.

2.1 Structural Equation Model

Given the graph \mathcal{G} , the covariates of all units in the graph \mathbf{X} , and the entire treatment assignments vector \mathbf{T} , the structural equation model describing the considered data generation process is given as follows

$$\begin{aligned} T_i &= f_T(X_i) \\ Y_i &= f_Y(T_i, \mathbf{X}, \mathbf{T}, \mathcal{G}) + \epsilon_{Y_i}, \end{aligned} \quad (2)$$

for units $i = 1, \dots, n$. This structural equation model encodes both the observational studies and the randomized experiments setting. In observational studies, e.g., on the Amazon dataset (see Section 4.1), the treatment \mathbf{T}_i depends on the covariate \mathbf{X}_i and the unknown specification of f_T , or even on the neighboring units under network interference. In the setting of the randomized experiment, e.g., experiments on Wave1 and Pokec datasets, the treatment assignment function is specified as $f_T = \text{Bern}(p)$, where p represents predefined treatment probability. Function f_Y characterizes the causal response, which depends on, in addition to \mathbf{X}_i and \mathbf{T}_i , the graph and neighboring covariates and treatment assignments. If only influences from first-order neighbors are considered, the response generation can be specified as $Y_i = f_Y(T_i, \mathbf{X}_{\mathcal{N}_i}, \mathbf{T}_{\mathcal{N}_i}, \mathcal{G}) + \epsilon_{Y_i}$. When the graph structure is given and fixed, we leave out \mathcal{G} in the notation.

2.2 Distribution Discrepancy Penalty

Even without network interference, a covariate shift problem of counterfactual inference is commonly observed, namely the factual distribution $\Pr(\mathbf{X}, T)$ differs from the counterfactual distribution $\Pr(\mathbf{X}, 1 - T)$. To avoid biased inference, [18, 35] propose a balancing counterfactual inference using domain-adapted representation learning. Covariate vectors are first mapped to a feature space via a feature map Φ . In the feature space, treated and control populations are balanced by penalizing the distribution discrepancy between $\Pr(\Phi(\mathbf{X})|T = 0)$ and $\Pr(\Phi(\mathbf{X})|T = 1)$ using the *Integral Probability Metric*. This approach is equivalent to finding a feature space such that the treatment assignment T and representation $\Phi(\mathbf{X})$ become approximately disentangled, namely $\Pr(\Phi(\mathbf{X}), T) \approx \Pr(\Phi(\mathbf{X}))P(T)$. We use the Hilbert-Schmidt Independence Criterion (HSIC) as the dependence test in the feature space. The empirical HSIC using a Gaussian RBF kernel is written as $H\hat{S}IC_{\mathcal{K}_\sigma}$. According to [11], given samples $\{\Phi(\mathbf{X}_i), T_i\}_{i=1}^n$, the empirical estimation of HSIC in Gaussian kernel \mathcal{K}_σ reads

$$\begin{aligned} H\hat{S}IC_{\mathcal{K}_\sigma} &= \frac{1}{n^2} \sum_{i,j=1}^n \mathcal{K}_\sigma(\Phi(\mathbf{X}_i), \Phi(\mathbf{X}_j)) \mathcal{K}_\sigma(T_i, T_j) \\ &+ \frac{1}{n^4} \sum_{i,j,k,l=1}^n \mathcal{K}_\sigma(\Phi(\mathbf{X}_i), \Phi(\mathbf{X}_j)) \mathcal{K}_\sigma(T_k, T_l) - \frac{2}{n^3} \sum_{i,j,k=1}^n \mathcal{K}_\sigma(\Phi(\mathbf{X}_i), \Phi(\mathbf{X}_j)) \mathcal{K}_\sigma(T_i, T_k). \end{aligned}$$

Note that incorporating the feature map and the representation balancing penalty is essential to tackle the imbalanced assignments in observational studies, e.g., on the Amazon dataset (see Section 4.1).

2.3 Graph Neural Networks

Different GNNs are employed and compared in our model, and we briefly provide a review.

Graph Convolutional Network (GCN) [21] The graph convolutional layer in GCN is defined as

$$\mathbf{X}^{(l+1)} = \sigma \left(\hat{\mathbf{D}}^{-1/2} \hat{\mathbf{A}} \hat{\mathbf{D}}^{-1/2} \mathbf{X}^{(l)} \mathbf{W}^{(l)} \right),$$

where $\mathbf{X}^{(l+1)}$ is the hidden output from the l -th layer with $\mathbf{X}^{(0)}$ being the input features matrix, and σ is the activation function, e.g., ReLU. The modified adjacency $\hat{\mathbf{A}}$ with inserted self-connections is defined as $\hat{\mathbf{A}} := \mathbf{A} + \mathbf{I}$, and $\hat{\mathbf{D}}$ denotes the node degree matrix of $\hat{\mathbf{A}}$.

GraphSAGE GraphSAGE [12] is an inductive framework for calculating node embeddings and aggregating neighbor information. The mean aggregation operator of the GraphSAGE in this work reads

$$\mathbf{X}_i^{(l+1)} = \text{norm} \left(\text{mean}_{j \in \mathcal{N}_i \cup \{i\}} \mathbf{X}_j^{(l)} \mathbf{W}^{(l)} \right),$$

with norm being the normalization operator. Traditional GCN algorithms perform spectral convolution via eigen-decomposition of the full graph Laplacian. In contrast, GraphSAGE computes a localized convolution by aggregating the neighborhood around a node, which resembles the simulation protocol of linear treatment response with spillover effect for semi-synthetic experiments (see Section 4.1). Due to the resemblance, a better causal estimator is expected when using GraphSAGE as the aggregation function (see Appendix F.3 for more heuristic motivations.).

1-GNN 1-GNN [28] is a variation of GraphSAGE, which performs separate transformations of node features and aggregated neighborhood features. Since the features of the considered unit and its neighbors contribute differently to the superimposed outcome, it is expected that the 1-GNN is more expressive than GraphSAGE. The convolutional operator of 1-GNN has the form

$$\mathbf{X}_i^{(l+1)} = \sigma \left(\mathbf{X}_i^{(l)} \mathbf{W}_1^{(l)} + \text{mean}_{j \in \mathcal{N}_i} \mathbf{X}_j \mathbf{W}_2^{(l)} \right). \quad (3)$$

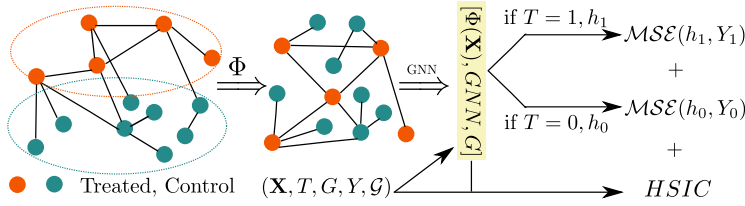


Fig. 1: Treated and control populations have different distributions in the covariate vectors space. Through a map Φ and distribution discrepancy term HSIC, features and treatment assignments become disentangled in the feature space. On top of Φ , we apply GNNs, where Φ and GNNs have 2 or 3 hidden layers, depending on the dataset. After applying GNNs, for each node i , the concatenation $[\Phi(\mathbf{X}_i), \text{GNN}(\Phi(\mathbf{X}), \mathbf{T})_i, G_i]$ is fed into outcome prediction network h_1 or h_0 depending on the treatment assignment. The loss function consists of outcome prediction error and the distribution discrepancy in the feature space.

2.4 GNN-based Causal Estimators

We use the percentage of treated neighboring nodes, i.e., the random variable G , as the treatment summary function, and the output of GNNs as the covariate aggregation function. The concatenation $[\Phi(\mathbf{X}_i), \text{GNN}(\Phi(\mathbf{X}), \mathbf{T})_i, G_i]$ of node i is then fed into the outcome prediction network h_1 or h_0 , depending on T_i , where h_1 and h_0 are neural networks with a scalar output. Note that $\text{GNN}(\Phi(\mathbf{X}), \mathbf{T})_i$ indicates that the treatment vector \mathbf{T} is also

a GNNs' input. During the implementation, the treatment assignment vector masks the covariates, and GNN models use the masked covariates $T_i \mathbf{X}_i$, for $i = 1, \dots, n$, as inputs. In summary, given $(\Phi(\mathbf{X})_i, T_i, G_i, Y_i)$ and graph \mathcal{G} , the loss function for GNN-based estimators is defined as

$$\mathcal{L}_{\text{est}} := \text{MSE}(h_{T_i}([\Phi(\mathbf{X}_i), \text{GNN}(\Phi(\mathbf{X}), \mathbf{T})_i, G_i]), Y_i) + \kappa \hat{H\hat{S}IC}_{\mathcal{K}_\sigma},$$

where κ and σ are tunable hyperparameters. Our model is illustrated in Fig. 1. During the implementation, we incorporate two types of empirical representation balancing: balancing the outputs of representation network Φ to tackle imbalanced assignments, denoted as $\hat{H\hat{S}IC}^\Phi$, and balancing the outputs of the GNN representations to tackle imbalanced spillover exposure, denoted as $\hat{H\hat{S}IC}^{\text{GNN}}$.

At this point, it is necessary to emphasize that only the causal responses of a part of the units in \mathcal{N} are relevant to the models. The GNN-based models use this part of causal responses, the network structure \mathcal{G} , and covariates \mathbf{X} as input, and can predict the superimposed causal effects of the remaining units. Note that for GNN-based nonparametric models, the identifiability of causal response is guaranteed under reasonable assumptions similar to those given in Section 3.2 of [29]. The proof is relegated to Appendix A.

Notice that the outcome prediction networks h_0 and h_1 are trained to estimate the superposition of individual treatment effect and spillover effect. Still, after fitting the observed outcomes, we expect to extract the non-interfered individual treatment effect from the causal estimators by assuming that the considered unit is isolated. An individual treatment effect estimator can be defined similarly to Eq. 1. To be more specific, the individual treatment effect of unit i is expected to be extracted from GNN-based estimators by setting its exposure to $G_i = 0$ and its neighbors' covariates to $\mathbf{0}$, namely¹

$$\hat{\tau}(\mathbf{X}_i) = h_1([\Phi(\mathbf{X}_i), \mathbf{0}, 0]) - h_0([\Phi(\mathbf{X}_i), \mathbf{0}, 0]). \quad (4)$$

3 Intervention Policy on Graph

After obtaining the treatment effect estimator, we develop an algorithm for learning intervention assignments to maximize the utility on the entire graph; the learned rule for assignment is called a policy. As suggested in [3], without interference a utility function is defined as

$$A(\pi) = \mathbb{E}[(2\pi(\mathbf{X}_i) - 1)(Y_i(T_i = 1) - Y_i(T_i = 0))] = \mathbb{E}[(2\pi(\mathbf{X}_i) - 1)\tau(\mathbf{X}_i)].$$

An optimal policy $\hat{\pi}_n$ is obtained by maximizing the n -sample empirical utility function $\hat{A}_n^\tau(\pi) := \frac{1}{n} \sum_{i=1}^n (2\pi(\mathbf{X}_i) - 1)\hat{\tau}(\mathbf{X}_i)$ given the individual treatment response estimator $\hat{\tau}$, i.e., $\hat{\pi}_n \in \arg\max_{\pi \in \Pi} \hat{A}_n^\tau(\pi)$, where Π indicates the policy function class. Notably, $\hat{\pi}_n$ tends to assign treatment to units with positive treatment effect and control to units with negative responses.

Now, consider the outcome variable Y_i under network interference. For notational simplicity and clarity of the later proof, we assume first-order interference from nearest neighboring units, hence the outcome variable can be written as $Y_i(T_i, \mathbf{X}_{\mathcal{N}_i}, T_{\mathcal{N}_i})$. Inspired by the definition of $A(\pi)$, the utility function of a policy π under interference is defined as

$$S(\pi) := \mathbb{E}[(2\pi(\mathbf{X}_i) - 1)(Y_i(T_i = 1, \mathbf{X}_{\mathcal{N}_i}, T_{\mathcal{N}_i} = \pi(\mathbf{X}_{\mathcal{N}_i})) - Y_i(T_i = 0, \mathcal{G} = \emptyset))], \quad (5)$$

¹ Spillover effect can be extracted similarly.

where $Y_i(T_i = 0, \mathcal{G} = \emptyset)$ with an empty graph represents the individual outcome under control without any network influence.² After some manipulations, $S(\pi)$ equals the sum of individual treatment effect and spillover effect, i.e.,

$$S(\pi) = \mathbb{E}[(2\pi(\mathbf{X}_i) - 1)(\tau_i + \delta_i(\pi))],$$

where

$$\begin{aligned}\tau_i &:= \mathbb{E}[Y_i(T_i = 1, \mathcal{G} = \emptyset) - Y_i(T_i = 0, \mathcal{G} = \emptyset) | \mathbf{X}_i] \\ \delta_i(\pi) &:= \mathbb{E}[Y_i(T_i = 1, \mathbf{X}_{\mathcal{N}_i}, T_{\mathcal{N}_i} = \pi(\mathbf{X}_{\mathcal{N}_i})) - Y_i(T_i = 1, \mathcal{G} = \emptyset) | \mathbf{X}_i, \mathbf{X}_{\mathcal{N}_i}].\end{aligned}$$

To be more specific, τ_i is the conventional individual treatment effect, while $\delta_i(\pi)$ represents the spillover effect under the policy π and when $T_i = 1$. Due to the network-dependency in the spillover effect, an optimal policy will not merely treat units with positive individual treatment effect but also adjust its intervention on the entire graph to maximize the spillover effects.

Next, we establish guarantees for the regret of learned intervention policy. Let $\hat{\tau}_i$ and $\hat{\delta}_i(\pi)$ denote the estimator of τ_i and $\delta_i(\pi)$, respectively. Given the true models τ_i and $\delta_i(\pi)$, let

$$S_n^{\pi, \delta}(\pi) := \frac{1}{n} \sum_{i=1}^n (2\pi(\mathbf{X}_i) - 1)(\tau_i + \delta_i(\pi))$$

be the empirical analogue of $S(\pi)$, and let

$$\hat{S}_n^{\pi, \delta}(\pi) := \frac{1}{n} \sum_{i=1}^n (2\pi(\mathbf{X}_i) - 1)(\hat{\tau}_i + \hat{\delta}_i(\pi)) \quad (6)$$

be the empirical utility with estimators plugged in. Using learned causal estimators, an optimal intervention policy from the empirical utility perspective can be obtained from $\hat{\pi}_n \in \arg\max_{\pi \in \Pi} \hat{S}_n^{\pi, \delta}(\pi)$. Moreover, the best possible intervention policy from the functional class Π with respect to the utility $S(\pi)$ is written as $\pi^* := \arg\max_{\pi \in \Pi} S(\pi)$, and the policy regret between π^* and $\hat{\pi}_n$ is defined as

$$\mathcal{R}(\hat{\pi}_n) := S(\pi^*) - S(\hat{\pi}_n).$$

Throughout the estimation of policy regret, we maintain the following assumptions.

Assumption 1.

(BO) *Bounded treatment and spillover effects:* There exist $0 < M_1, M_2 < \infty$ such that the individual treatment effect satisfies $|\tau_i| \leq M_1$ and the spillover effect satisfies $\forall \pi \in \Pi, |\delta_i(\pi)| \leq M_2$.

(WI) *Weak independence assumption:* For any node indices i and j , the weak independence assumption assumes that $\mathbf{X}_i \perp \mathbf{X}_j$ if $A_{ij} = 0$, or $\nexists k$ with $A_{ik} = A_{kj} = 1$.

(LIP) *Lipschitz continuity of the spillover effect w.r.t. policy:* Given two treatment policies π_1 and π_2 , for any node i the spillover effect satisfies $|\delta_i(\pi_1) - \delta_i(\pi_2)| \leq L \|\pi_1 - \pi_2\|_\infty$, where the Lipschitz constant satisfies $L > 0$ and $\|\pi_1 - \pi_2\|_\infty := \sup_{\mathbf{X} \in \mathcal{X}} |\pi_1(\mathbf{X}) - \pi_2(\mathbf{X})|$.

(ES) *Uniformly consistency:* after fitting experimental or observational data on \mathcal{G} , individual treatment effect estimator satisfies

$$\frac{1}{n} \sum_{i=1}^n |\tau_i - \hat{\tau}_i| < \frac{\alpha_\tau}{n^{\zeta_\tau}},$$

² Hence $\mathbf{X}_{\mathcal{N}_i}$ and $T_{\mathcal{N}_i}$ are omitted in the expression.

and spillover estimator satisfies

$$\forall \pi \in \Pi, \frac{1}{n} \sum_{i=1}^n |\delta_i(\pi) - \hat{\delta}_i(\pi)| < \frac{\alpha_\delta}{n\zeta_\delta} \quad (7)$$

where $\alpha_\tau > 0$ and $\alpha_\delta > 0$ are scaling factors that characterize the errors of estimators. ζ_τ and ζ_δ control the convergence rate of estimators for individual treatment effect and spillover effect, respectively, which satisfy $0 < \zeta_\tau, \zeta_\delta < 1$.

Notice that the (ES) assumption requires consistent estimators of the individual treatment effect and the spillover effect, which is the fundamental problem of causal inference with interference. In our GNN-based model, these empirical errors are particularly difficult to estimate due to the lack of proper theoretical tools for understanding GNNs. To grasp how these GNN-based causal estimators are influenced by the network structure and network effect, in Appendix F.3, we study a particular class of GNNs, which is inspired by the *surrogate model* of nonlinear graph neural networks and have the following claim.

Claim 1. *GNN-based causal estimators restricted to a particular class for predicting the superimposed causal effects have an error bound $\mathcal{O}(\sqrt{\frac{D_{max}^3 \ln D_{max}}{n}})$, where $D_{max} := 1 + d_{max} + d_{max}^2$ and d_{max} is the maximal node degree in the graph.*

The above claim indicates that an accurate and consistent causal estimator is difficult with large network effects. Worse case is that the $\frac{1}{\sqrt{n}}$ convergence rate in the (ES) assumption becomes unreachable when $d_{max}(n)$ depends on the number of units. The exact convergence rate of causal estimators is impossible to derive since it depends on the topology of the network, and it beyond the theoretical scope of this work. Therefore, we assume the coefficients ζ_τ and ζ_δ to characterize the convergence rates, which is line with the assumption made in [3] (see Assumption 2 of [3]).

Besides, (LIP) assumes that the change of received spillover effect is bounded after modifying the treatment assignments of one unit's neighbors. We will use hypergraph techniques, instead of chromatic number arguments, to give a tighter bound of policy regrets. Another advantage is that the weak independence (WI) assumption can be relaxed to support longer dependencies on the network. However, by relaxing (WI), the power of d_{max} in Theorem 1 and 5 needs to be modified correspondingly. For example, if we assume a next-nearest neighbors dependency of covariates, i.e., $\mathbf{X}_i \perp \mathbf{X}_j$ for $j \notin i \cup \mathcal{N}_i \cup \mathcal{N}_i^{(2)}$, then the term d_{max}^2 in Theorem 1 and 5 needs to be modified to d_{max}^4 .

Under Assumption 3, we can have the following bound.

Theorem 1. *By Assumption 3, for any small $\epsilon > 0$, the policy regret is bounded by $\mathcal{R}(\hat{\pi}_n) \leq 2\left(\frac{\alpha_\tau}{n\zeta_\tau} + \frac{\alpha_\delta}{n\zeta_\delta}\right) + 2\epsilon$ with probability at least*

$$1 - \mathcal{N}\left(\Pi, \frac{\epsilon}{4(2M_1 + 2M_2 + L)}\right) \exp\left(-\frac{n\epsilon^2}{32(d_{max}^2 + 1)(M_1 + M_2)^2}\right)$$

where $\mathcal{N}\left(\Pi, \frac{\epsilon}{4(2M_1 + 2M_2 + L)}\right)$ indicates the covering number³ on the functional class Π with radius $\frac{\epsilon}{4(2M_1 + 2M_2 + L)}$, and d_{max} is the maximal node degree in the graph \mathcal{G} .

³ The covering number characterizes the capacity of a functional class. Definition is provided in the Appendix F.1

Proof. Under (WI) and (BO), we can use concentration inequalities of networked random variables defined on a hypergraph, which is derived from graph \mathcal{G} to bound the convergence rate. Moreover, the Lipschitz assumption (LIP) allows an estimation of the covering number of the policy functional class Π (see Appendix F.1). ■

Suppose that the policy functional class Π is finite and its capacity is bounded by $|\Pi|$. According to Theorem 1, with probability at least $1 - \delta$, the policy regret is bounded by

$$\begin{aligned} \mathcal{R}(\hat{\pi}_n) &\leq 2 \left(\frac{\alpha_\tau}{n\zeta_\tau} + \frac{\alpha_\delta}{n\zeta_\delta} \right) + 8(M_1 + M_2) \sqrt{\frac{2(d_{\max}^2 + 1)}{n} \log \frac{|\Pi|}{\delta}} \\ &\approx 2 \left(\frac{\alpha_\tau}{n\zeta_\tau} + \frac{\alpha_\delta}{n\zeta_\delta} \right) + 8d_{\max}(M_1 + M_2) \sqrt{\frac{2}{n} \log \frac{|\Pi|}{\delta}} \end{aligned}$$

It indicates that optimal policies are more difficult to find in a dense graph even under weak interactions between neighboring nodes.

In a real-world setting, treatments could be expensive. So the policymaker usually encounters a budget or capacity constraints, e.g., the proportion of patients receiving treatment is limited, and to decide who should be treated under constraints is a challenging problem [22]. Through the interference-free welfare function $A(\pi)$, a policy is trained to make treatment choices using only each individual's features. In contrast, under interference, a smart policy should maximize the utility function Eq. (5) by deciding whether to treat an individual or expose it under neighboring treatment effects such that a required constraint can be satisfied. Therefore, in the second part of the experiments, after fitting causal estimators, we investigate policy networks that maximize the utility function $S(\pi)$ on the graph and satisfy a treatment proportion constraint.

To be more specific, we consider the constraint where only p_t percentage of the population can be assigned to treatment⁴. The corresponding sample-averaged loss function for a policy network π under capacity constraint is defined as

$$\mathcal{L}_{\text{pol}}(\pi) := -\hat{S}_n^{\tau, \delta}(\pi) + \gamma \left(\frac{1}{n} \sum_{i=1}^n \pi(\mathbf{X}_i) - p_t \right),$$

where γ is a hyperparameter for the constraint. Optimal policy under capacity constraint is obtained by

$$\hat{\pi}_n^{p_t} \in \min_{\pi \in \Pi} \mathcal{L}_{\text{pol}}(\pi).$$

A capacity-constrained policy regret bound is provided in Theorem 5, which is proved in Appendix F.2. It indicates that if, in the constraint, p_t is small, then the optimal capacity-constrained policy will be challenging to find. Increasing the treatment probability can not guarantee the improvement of the group's interest due to the non-linear network effect. Therefore, finding the balance between optimal treatment probability, treatment assignment, and group's welfare is a provocative question in social science.

Theorem 2. *By Assumption 3, for any small $\epsilon > 0$, the policy regret under the capacity constraint p_t is bounded by $\mathcal{R}(\hat{\pi}_n^{p_t}) \leq 2 \left(\frac{\alpha_\tau}{n\zeta_\tau} + \frac{\alpha_\delta}{n\zeta_\delta} \right) + 2\epsilon$ with probability at least $1 - \mathcal{N} \exp \left(-\frac{n\epsilon^2}{32(d_{\max}^2 + 1)(M_1 + M_2)^2} \right)$, where $\mathcal{N} := \mathcal{N} \left(\Pi, \frac{\epsilon}{8[(M_1 + M_2 + L) + \frac{1}{p_t}(M_1 + M_2)]} \right)$ indicates the covering number on the functional class Π with radius $\frac{\epsilon}{8[(M_1 + M_2 + L) + \frac{1}{p_t}(M_1 + M_2)]}$, and d_{\max} is the maximal node degree in the graph \mathcal{G} .*

⁴ Note that here p_t differs from the treatment probability p from causal structural equations in the randomized experiment setting.

4 Experiments

4.1 Datasets

The difficulties of evaluating the performance of the proposed estimators lie in the broad set of missing outcomes under counterfactual inference. Therefore, we conduct randomized experiments on two semi-synthetic datasets with *ground-truth* response generation functions, and observational studies on one real dataset with *unknown* treatment assignment and response generation functions. Notably, in the randomized experiment setting, we consider a linear response generation function inspired by Eq. 5 of [40],

$$G_0 : Y_i = Y_i(T_i = 0, \mathcal{G} = \emptyset) + T_i \tau(\mathbf{X}_i) + \delta_i(\mathbf{X}, \mathbf{T}, \mathcal{G}) + \epsilon_{Y_i},$$

where $Y_i(T_i = 0, \mathcal{G} = \emptyset)$ is the outcome under control and without network interference, and ϵ_{Y_i} represents Gaussian noise. $\tau(\mathbf{X}_i)$ and $\delta_i(\mathbf{X}, \mathbf{T}, \mathcal{G})$ represent individual treatment effect and spillover effect, respectively, whose forms are dataset-dependent and discussed below.

To further investigate the superiority of the GNN-based causal estimators on nonlinear causal responses, we consider the following data generation function inspired by Section 4.2 of [40],

$$G_1 : Y_i = Y_i(T_i = 0, \mathcal{G} = \emptyset) + T_i \tau(\mathbf{X}_i) + \delta_i(\mathbf{X}, \mathbf{T}, \mathcal{G}) + \kappa \delta_i^2(\mathbf{X}, \mathbf{T}, \mathcal{G}) + \epsilon_{Y_i},$$

where κ characterizes the strength of nonlinear effects. In addition, a more complicated nonlinear response generation function

$$G_2 : Y_i = Y_i(T_i = 0, \mathcal{G} = \emptyset) + T_i \tau(\mathbf{X}_i) + \delta_i(\mathbf{X}, \mathbf{T}, \mathcal{G}) + \frac{\kappa}{2} \delta_i^2(\mathbf{X}, \mathbf{T}, \mathcal{G}) + \frac{\kappa}{2} \tau(\mathbf{X}_i) \delta_i(\mathbf{X}, \mathbf{T}, \mathcal{G}) + \epsilon_{Y_i}$$

is considered, where the quadratic terms signify the spillover effect depending on the individual treatment effect.

Wave1 Wave1 is an in-school questionnaire data collected through the National Longitudinal Study of Adolescent Health project [6]. The questionnaire contains questions such as age, grade, health insurance, etc. Due to the anonymity of Wave1, we use the symmetrized k -NN graph derived from the questionnaire data as the friendship network. In our experiments, we choose $k = 10$, and the resulting friendship network has 5,578 nodes and 100,158 links. We assume a randomized experiment conducted on the friendship network which describes students' improvements of performance through assigning to a tutoring program or through the peer effect. Hence $Y_i(T_i = 0, \mathcal{G} = \emptyset)$ represents the overall performance of student i before assignment to a tutoring program and before being exposed to peer influences, $\tau(\mathbf{X}_i)$ the simulated performance difference after an assignment, and $\delta_i(\mathbf{X}, \mathbf{T}, \mathcal{G})$ the synthetic peer effect. Exact forms of $Y_i(T_i = 0, \mathcal{G} = \emptyset)$ and $\tau(\mathbf{X}_i)$ depend nonlinearly on the features of each student. Moreover, the first-order peer effect is simulated as

$$\delta_i(\mathbf{X}, \mathbf{T}, \mathcal{G}) := \alpha \frac{1}{|\mathcal{N}_i|} \sum_{j \in \mathcal{N}_i} T_j \tau(\mathbf{X}_j),$$

where the decay parameter α characterizes the decay of influence. In randomized experiments reported in the main text, we randomly assign 10% of the population to the treatment. Details of the generating process and more experiment results with different settings are relegated to Appendix B and E.

Pokec The friendship network derived from the Wave1 questionnaire data may violate the power-law degree distribution of real networks. Hence, we further conduct experiments on the real social network Pokec [37] with generated responses. Pokec is an online social network in Slovakia with profile data, including age, gender, education, etc. We consider randomized experiments on the Pokec social network, in which personalized advertisements of a new health medicine are pushed to some users. We assume that the response of exposed users to the advertisement only depends on a few properties, such as age, weight, smoking status, etc. We keep profiles with complete information on these properties, and the resulting Pokec social network contains 11,623 nodes and 76,752 links. Let $Y_i(T_i = 0, \mathcal{G} = \emptyset)$ represent the purchase of this new health medicine without external influence on the decision, $\tau(\mathbf{X}_i)$ the purchase difference after seeing the advertisement, $\delta_i(\mathbf{X}, \mathbf{T}, \mathcal{G})$ the purchase difference due to social influences. For randomized experiments on the Pokec social network, we also consider peer effects from next-nearest neighbors by defining

$$\delta_i(\mathbf{X}, \mathbf{T}, \mathcal{G}) := \alpha \frac{1}{|\mathcal{N}_i|} \sum_{j \in \mathcal{N}_i} T_j \tau(\mathbf{X}_j) + \alpha^2 \frac{1}{|\mathcal{N}_i^{(2)}|} \sum_{k \in \mathcal{N}_i^{(2)}} T_k \tau(\mathbf{X}_k),$$

where the decay parameter α characterizes the decay of influence. Details and more experimental results with different hyperparameter settings are given in Appendix C and E.

Amazon The co-purchase dataset from Amazon contains product details, review information, and a list of similar products. Therefore, there is a directed network of products that describes whether a substitutable or complementary product is getting co-purchased with another product [25]. To study the causal effect of reviews on the sales of products, [30] generates a dataset containing products with only positive reviews from the Amazon co-purchase dataset, named as pos Amazon, and Amazon for short. In this dataset, all items have positive reviews, i.e., the average rating is larger than 3, and one item is considered to be treated if there are more than three reviews under this item; otherwise, an item is in the control group. In this setting, pos Amazon is an over-treated dataset with more than 70% of products being in the treatment group. Word2vec embedding of an item's review serves as the feature vector of this item. Moreover, the individual treatment effect of an item is approximated by matching it to other items having similar features and under minimal exposure to neighboring nodes' treatments.

4.2 Results of Causal Estimators

Evaluation Metrics One evaluation metric is the square root of MSE for the prediction of the observed outcomes on the test dataset \mathcal{U}_T , which is defined as

$$\sqrt{\mathcal{MSE}} := \sqrt{\frac{1}{|\mathcal{U}_T|} \sum_{i \in \mathcal{U}_T} (Y_i - h_{T_i})^2},$$

where h_{T_i} denotes the output of the outcome prediction network (see h_0 and h_1 in Fig. 1). This metric reflects how well an estimator can predict the superimposed individual treatment and spillover effects on a network. Another evaluation metric that quantifies the quality of extracted individual treatment effect is the Precision in Estimation of Heterogeneous Effect studied in [14], which is defined as

$$\epsilon_{PEHE} := \frac{1}{|\mathcal{U}_T|} \sum_{i \in \mathcal{U}_T} (\tau(\mathbf{X}_i) - \hat{\tau}(\mathbf{X}_i))^2,$$

where $\hat{\tau}(\mathbf{X}_i)$ is defined in Eq. (4).

Table 1: Experimental results of randomized experiments on the Wave1 and Pokec datasets using linear response generation function G_0 . For Wave1, we set (node degree) $k = 10$, (decay parameter) $\alpha = 0.5$, and (treatment probability) $p = 0.1$, and for Pokec $p = 0.1$. Improvements are obtained by comparing with the best baselines.

	Wave1		Pokec	
	\sqrt{MSE}	ϵ_{PEHE}	\sqrt{MSE}	ϵ_{PEHE}
DA GB	0.721 ± 0.054	0.289 ± 0.061	0.713 ± 0.016	0.321 ± 0.057
DA RF	1.037 ± 0.122	0.790 ± 0.215	0.749 ± 0.023	0.840 ± 0.087
DR GB	0.831 ± 0.109	0.499 ± 0.185	0.686 ± 0.020	0.275 ± 0.051
DR EN	0.929 ± 0.091	0.733 ± 0.135	0.695 ± 0.019	0.247 ± 0.060
GPS	0.238 ± 0.012	0.150 ± 0.047	0.329 ± 0.010	0.147 ± 0.010
GCN + $H\hat{S}IC^{\Phi/GNN}$	0.192 ± 0.019	0.047 ± 0.018	0.305 ± 0.011	0.136 ± 0.009
GraphSAGE + $H\hat{S}IC^{\Phi/GNN}$	0.181 ± 0.016	0.042 ± 0.020	0.303 ± 0.008	0.123 ± 0.003
1-GNN + $H\hat{S}IC^{\Phi/GNN}$	0.176 ± 0.011	0.035 ± 0.011	0.302 ± 0.004	0.130 ± 0.006
Improve	26.1%	76.7%	8.2%	16.3%

Baselines Baseline models are domain adaption method [24] with gradient boosting regression (**DA GB**), with random forest regression (**DA RF**), doubly-robust estimator [10] with gradient boosting regression (**DR GB**), and elastic net regression (**DR EN**). They are implemented via EconML [31] with grid-searched hyperparameters. These baselines incorporate the feature vectors as inputs and exposure as the control variable into the model. For randomized experiments on Wave1 and Pokec, the predefined treatment probability p is provided, while for the observational studies on the Amazon dataset, the covariate-dependent treatment probability is estimated. Moreover, the generalized propensity score (GPS) method is reproduced and enhanced for a fair comparison, equipped with the same feature map Φ function. More details of baselines, the sketch of the training procedure, and hyperparameters are relegated to Appendix E.

Table 2: Experimental result on the pos Amazon dataset without representation balancing and under different imbalance penalties.

	\sqrt{MSE}	ϵ_{PEHE}
DA GB	0.601 ± 0.007	1.370 ± 0.016
DA RF	0.604 ± 0.019	1.398 ± 0.013
DR GB	0.615 ± 0.022	1.222 ± 0.020
DR EN	1.104 ± 0.001	1.929 ± 0.003
GPS	0.399 ± 0.003	1.968 ± 0.025
GCN	0.312 ± 0.002	2.400 ± 0.201
GCN + $H\hat{S}IC^{GNN}$	0.303 ± 0.006	1.881 ± 0.076
GCN + $H\hat{S}IC^{\Phi}$	0.301 ± 0.002	1.531 ± 0.024
GraphSAGE	0.305 ± 0.001	1.984 ± 0.026
GraphSAGE + $H\hat{S}IC^{GNN}$	0.296 ± 0.002	1.567 ± 0.051
GraphSAGE + $H\hat{S}IC^{\Phi}$	0.300 ± 0.002	1.358 ± 0.025
1-GNN	0.279 ± 0.000	1.512 ± 0.111
1-GNN + $H\hat{S}IC^{GNN}$	0.276 ± 0.002	1.434 ± 0.030
1-GNN + $H\hat{S}IC^{\Phi}$	0.277 ± 0.002	1.098 ± 0.031
Improve	30.8%	10.1%

Experiments We use partial outcomes, both in the randomized experiments and observational settings, to train the GNN-based causal estimators. We investigate the effect of penalizing representation imbalance in the observational studies on the Amazon dataset. The entire data points $(\mathbf{X}_i, T_i, G_i, Y_i)$ are randomly divided into training (80%), validation (5%), and test (15%) sets. Note that the entire network \mathcal{G} and the covariates of all units \mathbf{X} are given during the training and test, while only the causal responses of units in the training set are provided in the training phase. For the randomized experiments using the Wave1 and Pokec datasets, we repeat the experiments 3 times and use different random parameters in the response generation process each time.

Experimental results on the Wave1 and Pokec data generated via linear model G_0 are presented in Table 1. Both representation balancing $H\hat{S}IC^\Phi$ and $H\hat{S}IC^{GNN}$ are deployed in the GNN-based estimators for searching for the best performance. GNN-based estimators, especially the 1-GNN estimator, are superior for superimposed causal effects prediction. One can observe a 26.1% improvement of the \sqrt{MSE} metric on the Wave1 dataset when comparing the 1-GNN estimator with the enhanced GPS method and a 8.2% improvement on the Pokec dataset. The covariates of neighboring units in the Pokec dataset actually have strong cosine similarity, hence the improvement on the Pokec dataset is not significant, and the network effect can be approximately captured from the exposure variable. Table 2 shows the experimental results on the pos Amazon dataset in the observational study. In particular, we demonstrate the effects of without representation penalty, and with different penalties. It shows that representation penalties can significantly improve the individual treatment effect recovery, serving as a regularization to avoid over-fitting the network interference. Furthermore, GNN-based estimators using $H\hat{S}IC^{GNN}$ penalty are slightly better than those using $H\hat{S}IC^\Phi$ penalty; however, by sacrificing the metric ϵ_{PEHE} .

Table 3: Experimental results of randomized experiments on the Wave1 dataset using non-linear response generation functions G_1 and G_2 with $\kappa = 0.2$. For Wave1, we set (node degree) $k = 10$, (decay parameter) $\alpha = 0.5$, and (treatment probability) $p = 0.1$. $H\hat{S}IC^\Phi$ and $H\hat{S}IC^{GNN}$ are deployed in the GNN-based estimators. Improvements are obtained by comparing with the best baselines.

	Wave1			
	G_1		G_2	
	\sqrt{MSE}	ϵ_{PEHE}	\sqrt{MSE}	ϵ_{PEHE}
DA GB	0.770 \pm .017	0.379 \pm .126	0.763 \pm .047	0.248 \pm .121
DA RF	1.047 \pm .046	0.701 \pm .029	0.977 \pm .021	0.599 \pm .193
DR GB	0.814 \pm .058	0.392 \pm .029	0.771 \pm .014	0.401 \pm .028
DR EN	1.063 \pm .037	0.843 \pm .005	0.886 \pm .010	0.636 \pm .173
GPS	0.236 \pm .001	0.158 \pm .031	0.262 \pm .071	0.163 \pm .063
GCN	0.192 \pm .003	0.050 \pm .007	0.201 \pm .034	0.044 \pm .026
GraphSAGE	0.191 \pm .004	0.049 \pm .003	0.198 \pm .022	0.039 \pm .018
1-GNN	0.207 \pm .003	0.058 \pm .006	0.188 \pm .020	0.043 \pm .024
Improve	19.1%	19.0%	28.2%	76.1%

Table 3 and 4 report the performance of GNN-based causal estimators on nonlinear response models. Nonlinear responses are generated via G_1 and G_2 under $\kappa = 0.2$. For the \sqrt{MSE} metric, GNN-based estimators outperform the best baseline GPS dramatically, showing the effectiveness of predicting nonlinear causal responses. Moreover, a 19.0%(G_1)

Table 4: Experimental results of randomized experiments on the Pokec dataset using nonlinear response generation functions G_1 and G_2 with $\kappa = 0.2$. For Pokec, we set $p = 0.1$. $H\hat{SIC}^\Phi$ and $H\hat{SIC}^{GNN}$ are deployed in the GNN-based estimators. Improvements are obtained by comparing with the best baselines.

	Pokec			
	G_1		G_2	
	\sqrt{MSE}	ϵ_{PEHE}	\sqrt{MSE}	ϵ_{PEHE}
DA GB	$0.988 \pm .005$	$0.419 \pm .046$	$1.189 \pm .017$	$0.376 \pm .033$
DA RF	$1.016 \pm .024$	$1.075 \pm .031$	$1.225 \pm .009$	$1.016 \pm .037$
DR GB	$0.943 \pm .024$	$0.297 \pm .057$	$1.173 \pm .012$	$0.314 \pm .020$
DR EN	$0.947 \pm .023$	$0.181 \pm .031$	$1.172 \pm .013$	$0.282 \pm .041$
GPS	$0.420 \pm .006$	$0.212 \pm .070$	$0.475 \pm .004$	$0.220 \pm .013$
GCN	$0.367 \pm .005$	$0.162 \pm .004$	$0.423 \pm .017$	$0.183 \pm .010$
GraphSAGE	$0.360 \pm .000$	$0.146 \pm .001$	$0.425 \pm .018$	$0.167 \pm .005$
1-GNN	$0.366 \pm .013$	$0.151 \pm .006$	$0.408 \pm .009$	$0.158 \pm .004$
Improve	14.3%	19.3%	14.1%	28.2%

and 76.1%(G_2) performance improvement on the ϵ_{PEHE} metric with the Wave1 dataset shows that setting an empty graph, i.e., $\mathcal{G} = \emptyset$, in the GNN-based estimators is an appropriate approach for extracting individual causal effect. Results of nonlinear responses with larger strength parameter $\kappa = 0.5$ are reported in Appendix B and C.

4.3 Results on Improved Intervention Policy

Experiment Settings After obtaining the optimal causal effect estimators and feature map Φ (see Fig. 1), we subsequently optimize intervention policy on the same graph. A neural network having two hidden layers, with ReLU activation between hidden layers and sigmoid activation at the end, is employed as the policy network. The output of the policy network lies in $[0, 1]$, and it is interpreted as the probability of treating a node. The real intervention choice is then sampled from this probability via the Gumbel-softmax trick [16] such that gradients can be back-propagated. Sampled treatment choices along with corresponding node features are then fed into the feature map Φ and subsequent causal estimators to evaluate the utility function under network interference defined in Eq. (5). Each experiment setting is repeated 5 times until convergence. The hyperparameter γ in \mathcal{L}_{pol} is tuned such that the constraint for the percentage p_t is satisfied within the tolerance ± 0.01 . More details of experiment settings and hyperparameters are relegated to Appendix E and D.

To quantify the optimized policy $\hat{\pi}_n^{p_t}$, we evaluate the difference

$$\Delta\hat{S}(\hat{\pi}_n^{p_t}) := \hat{S}_n^{\tau, \delta}(\hat{\pi}_n^{p_t}) - \hat{S}_n^{\tau, \delta}(\pi_R^{p_t}),$$

where $\pi_R^{p_t}$ represents a randomized intervention underlying the same capacity constraint. The difference $\Delta\hat{S}(\hat{\pi}_n^{p_t})$ indicates how a learned policy can outperform a randomized policy with the same constraint evaluated via learned causal effect estimators. However, from its definition, it is concerned that the policy improvement $\hat{\pi}_n^{p_t}$ may be very biased, such that any “expected improvement” may come from the inaccurate causal estimators. Hence, for the Wave1 and Pokec datasets, knowing the generating process of treatment and spillover effects, we also compare the actual utility difference

$$\Delta S(\hat{\pi}_n^{p_t}) := S_n^{\tau, \delta}(\hat{\pi}_n^{p_t}) - S_n^{\tau, \delta}(\pi_R^{p_t}).$$

Table 5: Intervention policy improvements on the Wave1 and Pokec semi-synthetic datasets under treatment capacity constraint with $p_t = 0.3$. $\Delta\hat{S}(\hat{\pi}_n^{p_t})$ and $\Delta S(\hat{\pi}_n^{p_t})$ represent utility differences evaluated from learned estimators and ground truth, respectively. Note that *only* $\Delta S(\hat{\pi}_n^{p_t})$ reflects the real policy improvement.

	Wave1		Pokec	
	$\Delta\hat{S}(\hat{\pi}_n^{p_t})$	$\Delta S(\hat{\pi}_n^{p_t})$	$\Delta\hat{S}(\hat{\pi}_n^{p_t})$	$\Delta S(\hat{\pi}_n^{p_t})$
DA GB	0.276 ± 0.033	0.002 ± 0.025	0.231 ± 0.051	0.001 ± 0.036
DA RF	0.302 ± 0.029	0.003 ± 0.021	0.198 ± 0.080	0.001 ± 0.057
DR GB	0.322 ± 0.023	0.002 ± 0.019	0.338 ± 0.060	0.002 ± 0.046
DR EN	0.311 ± 0.019	0.001 ± 0.018	0.329 ± 0.028	0.001 ± 0.026
GPS	0.235 ± 0.042	0.004 ± 0.032	0.362 ± 0.069	0.001 ± 0.053
GCN	0.260 ± 0.024	0.163 ± 0.020	0.270 ± 0.007	0.190 ± 0.012
GraphSAGE	0.283 ± 0.031	0.176 ± 0.025	0.376 ± 0.049	0.211 ± 0.034
1-GNN	0.327 ± 0.038	0.208 ± 0.026	0.377 ± 0.041	0.225 ± 0.031

Table 6: Intervention policy improvements on the pos Amazon dataset under treatment capacity constraint with $p_t = 0.5$. Only domain adaption methods and GPS are compared since they are the best baseline estimators according to Table 2.

	DA GB	DA RF	GPS	GCN	GraphSAGE	1-GNN
$\Delta\hat{S}(\hat{\pi}_n^{p_t})$	38.9 ± 1.1	84.1 ± 2.3	98.6 ± 10.8	80.7 ± 0.9	86.0 ± 0.9	84.1 ± 1.3

Table 5 displays policy optimization results on the under-treated Wave1 and Pokec simulation datasets, where initially only 10% of nodes are randomly assigned to treatment. It shows that an optimized policy network cannot even outperform a randomized policy in ground truth when the causal estimators perform poorly. Hence, policy networks learned from the utility function with plugged in doubly-robust or domain adaption estimators are not reliable. By contrast, the small difference between genuine utility improvement $\Delta S(\hat{\pi}_n^{p_t})$ and estimated improvement $\Delta\hat{S}(\hat{\pi}_n^{p_t})$ for the GNN-based causal estimators indicates the reliability of the optimized policy. Moreover, comparing the ground-truth utility improvement on GPS and GCN-based estimator shows that the policy network sensitively relies on the accuracy of the employed causal estimator. Furthermore, one might argue that through baseline estimators, a simple policy network cannot adjust its treatment choice according to neighboring nodes' features and responses, unlike through GNN-based estimators. For a fair comparison, in Appendix D, we also provide experimental results using a GNN-based policy network. However, we still cannot observe genuine utility improvements on $\Delta S(\hat{\pi}_n^{p_t})$ when using baseline models as causal estimators.

Next, we conduct experiments for intervention policy learning on the over-treated pos Amazon dataset under treatment capacity constraint. Since we do not have access to the ground truth of the pos Amazon dataset, Table 6 shows the utility difference under treatment capacity constraint with $p_t = 0.5$ evaluated only from learned causal estimators. Although the optimized utility improvement $\Delta\hat{S}(\hat{\pi}_n^{p_t})$ achieves the best result via the GPS causal estimator, it might be unreliable compared to the ground truth. A reliable policy improvement having comparable utility improvement via a GNN-based causal estimator is expected.

5 Conclusion

In this work, we first introduced the task of causal inference under general network interference and proposed causal effect estimators using GNNs of various types. We also defined a novel utility function for policy optimization on interconnected nodes, of which a graph-dependent policy regret bound can be derived. We conduct experiments on semi-synthetic simulation and real datasets. Experiment results show that GNN-based causal effect estimators, especially GraphSAGE and 1-GNN, with an HSIC distribution discrepancy penalty, are superior in superimposed causal effect prediction, and the individual treatment effect can be recovered reasonably well. Subsequent experiments of intervention policy optimization under capacity constraint further confirms the importance of employing an optimal and reliable causal estimator for policy improvement. In future work, we consider the scenario in which the network structure is only partially observed, or dynamic.

Conflict of interest

The authors declare that they have no conflict of interest.

References

1. David Arbour, Dan Garant, and David Jensen. Inferring network effects from observational data. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 715–724. ACM, 2016.
2. Peter M Aronow, Cyrus Samii, et al. Estimating average causal effects under general interference, with application to a social network experiment. *The Annals of Applied Statistics*, 11(4):1912–1947, 2017.
3. Susan Athey and Stefan Wager. Efficient policy learning. *arXiv preprint arXiv:1702.02896*, 2017.
4. Peter L Bartlett, Olivier Bousquet, Shahar Mendelson, et al. Local rademacher complexities. *The Annals of Statistics*, 33(4):1497–1537, 2005.
5. Jake Bowers, Mark M. Fredrickson, and Costas Panagopoulos. Reasoning about interference between units: A general framework. *Political Analysis*, 21:97–124, 2013.
6. Kim Chantala and Joyce Tabor. National longitudinal study of adolescent health: Strategies to perform a design-based analysis using the add health data. 1999.
7. Felipe Cucker and Ding Xuan Zhou. *Learning theory: an approximation theory viewpoint*, volume 24. Cambridge University Press, 2007.
8. Luc Devroye, László Györfi, and Gábor Lugosi. *A probabilistic theory of pattern recognition*, volume 31. Springer Science & Business Media, 2013.
9. Laura Forastiere, Edoardo M Airoidi, and Fabrizia Mealli. Identification and estimation of treatment and interference effects in observational studies on networks. *arXiv preprint arXiv:1609.06245*, 2016.
10. Michele Jonsson Funk, Daniel Westreich, Chris Wiesen, Til Stürmer, M Alan Brookhart, and Marie Davidian. Doubly robust estimation of causal effects. *American journal of epidemiology*, 173(7):761–767, 2011.
11. Arthur Gretton, Olivier Bousquet, Alex Smola, and Bernhard Schölkopf. Measuring statistical dependence with hilbert-schmidt norms. In *International conference on algorithmic learning theory*, pages 63–77. Springer, 2005.

12. Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems*, pages 1024–1034, 2017.
13. Kathleen Mullan Harris and J Richard Udry. National longitudinal study of adolescent to adult health (add health), 1994–2008 [public use]. *Ann Arbor, MI: Carolina Population Center, University of North Carolina-Chapel Hill [distributor], Inter-university Consortium for Political and Social Research [distributor]*, pages 08–06, 2018.
14. Jennifer L Hill. Bayesian nonparametric modeling for causal inference. *Journal of Computational and Graphical Statistics*, 20(1):217–240, 2011.
15. Michael G. Hudgens and M. Elizabeth Halloran. Toward causal inference with interference. 103(482), 2008.
16. Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *International Conference on Learning Representations (ICLR)*, 2017.
17. Svante Janson. Large deviations for sums of partly dependent random variables. *Random Structures & Algorithms*, 24(3):234–248, 2004.
18. Fredrik Johansson, Uri Shalit, and David Sontag. Learning representations for counterfactual inference. In *International conference on machine learning*, pages 3020–3029, 2016.
19. Nathan Kallus. Balanced policy evaluation and learning. In *Advances in Neural Information Processing Systems*, pages 8895–8906, 2018.
20. Nathan Kallus and Angela Zhou. Confounding-robust policy improvement. In *Advances in Neural Information Processing Systems*, pages 9269–9279, 2018.
21. Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *International Conference on Learning Representations (ICLR)*, 2017.
22. Toru Kitagawa and Aleksey Tetenov. Who should be treated? empirical welfare maximization methods for treatment choice. Technical report, Cemmap working paper, 2017.
23. Toru Kitagawa and Aleksey Tetenov. Who should be treated? empirical welfare maximization methods for treatment choice. *Econometrica*, 86(2):591–616, 2018.
24. Sören R Künzle, Jasjeet S Sekhon, Peter J Bickel, and Bin Yu. Metalearners for estimating heterogeneous treatment effects using machine learning. *Proceedings of the National Academy of Sciences*, 116(10):4156–4165, 2019.
25. Jure Leskovec, Lada A Adamic, and Bernardo A Huberman. The dynamics of viral marketing. *ACM Transactions on the Web (TWEB)*, 1(1):5, 2007.
26. Lan Liu and Michael G Hudgens. Large sample randomization inference of causal effects in the presence of interference. *Journal of the american statistical association*, 109(505):288–301, 2014.
27. Charles F Manski. *Identification for prediction and decision*. Harvard University Press, 2009.
28. Christopher Morris, Martin Ritzert, Matthias Fey, William L Hamilton, Jan Eric Lenssen, Gaurav Rattan, and Martin Grohe. Weisfeiler and leman go neural: Higher-order graph neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 4602–4609, 2019.
29. Elizabeth L Ogburn, Oleg Sofrygin, Ivan Diaz, and Mark J van der Laan. Causal inference for social network data. *arXiv preprint arXiv:1705.08527*, 2017.
30. Vineeth Rakesh, Ruocheng Guo, Raha Moraffah, Nitin Agarwal, and Huan Liu. Linked causal variational autoencoder for inferring paired spillover effects. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*,

- pages 1679–1682. ACM, 2018.
31. Microsoft Research. EconML: A Python Package for ML-Based Heterogeneous Treatment Effects Estimation. <https://github.com/microsoft/EconML>, 2019. Version 0.x.
 32. Donald B Rubin. Estimating causal effects of treatments in randomized and nonrandomized studies. *Journal of educational Psychology*, 66(5):688, 1974.
 33. Donald B Rubin. Randomization analysis of experimental data: The fisher randomization test comment. *Journal of the American Statistical Association*, 75(371):591–593, 1980.
 34. Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. Computational capabilities of graph neural networks. *IEEE Transactions on Neural Networks*, 20(1):81–102, 2008.
 35. Uri Shalit, Fredrik D Johansson, and David Sontag. Estimating individual treatment effect: generalization bounds and algorithms. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 3076–3085. JMLR. org, 2017.
 36. Jerzy Splawa-Neyman, Dorota M Dabrowska, and TP Speed. On the application of probability theory to agricultural experiments. essay on principles. section 9. *Statistical Science*, pages 465–472, 1990.
 37. Lubos Takac and Michal Zabovsky. Data analysis in public social networks. In *International Scientific Conference and International Workshop Present Day Trends of Innovations*, volume 1, 2012.
 38. Eric J Tchetgen Tchetgen, Isabel Fulcher, and Ilya Shpitser. Auto-g-computation of causal effects on a network. *arXiv preprint arXiv:1709.01577*, 2017.
 39. Eric J Tchetgen Tchetgen and Tyler J VanderWeele. On causal inference in the presence of interference. *Statistical methods in medical research*, 21(1):55–75, 2012.
 40. Panos Toulis and Edward Kao. Estimation of causal peer influence effects. In *International conference on machine learning*, pages 1489–1497, 2013.
 41. Aad W Van Der Vaart and Jon A Wellner. Weak convergence. In *Weak convergence and empirical processes*, pages 16–28. Springer, 1996.
 42. Davide Viviano. Policy targeting under network interference. *arXiv preprint arXiv:1906.10258*, 2019.
 43. Martin J Wainwright. *High-dimensional statistics: A non-asymptotic viewpoint*, volume 48. Cambridge University Press, 2019.
 44. Yuyi Wang, Zheng-Chu Guo, and Jan Ramon. Learning from Networked Examples. In *28th International Conference on Algorithmic Learning Theory (ALT), Kyoto, Japan, October 2017*.
 45. Daniel Zügner, Amir Akbarnejad, and Stephan Günnemann. Adversarial attacks on neural networks for graph data. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2847–2856. ACM, 2018.

A Nonparametric Identifiability of Causal Effect

The nonparametric identifiability of expected causal response is guaranteed following [29, 9]. For the sake of simplicity, we assume that influences are only from the first-order neighbors. To prove the identifiability, we introduce a variable $V_i := S_{V,i}(\mathbf{X}_{\mathcal{N}_i}, \mathbf{T}_{\mathcal{N}_i})$, where

$$S_{V,i} : \{0, 1\}^{|\mathcal{N}_i|} \otimes \mathcal{X}^{\otimes |\mathcal{N}_i|} \rightarrow \mathcal{V}_i,$$

for $i = 1, \dots, n$, represents the aggregation of neighboring covariates and treatment assignments, e.g., the average of neighboring treatments and the output of a GNN. Following reasonable assumptions are necessary for the nonparametric identifiability.

Assumption 2.

(1) Given summary function $S_{V,i}$, for $i = 1, \dots, n$, $\forall \mathbf{T}_{\mathcal{N}_i}, \mathbf{T}'_{\mathcal{N}_i}, \forall \mathbf{X}_{\mathcal{N}_i}, \mathbf{X}'_{\mathcal{N}_i}, \forall \mathbf{T}_{\mathcal{N}_{-i}}, \mathbf{T}'_{\mathcal{N}_{-i}}$, and $\forall \mathbf{X}_{\mathcal{N}_{-i}}, \mathbf{X}'_{\mathcal{N}_{-i}}$, with $S_{V,i}(\mathbf{T}_{\mathcal{N}_i}, \mathbf{X}_{\mathcal{N}_i}) = S_{V,i}(\mathbf{T}'_{\mathcal{N}_i}, \mathbf{X}'_{\mathcal{N}_i})$, then it holds

$$Y_i(T_i, \mathbf{T}_{\mathcal{N}_i}, \mathbf{X}_{\mathcal{N}_i}, \mathbf{T}_{\mathcal{N}_{-i}}, \mathbf{X}_{\mathcal{N}_{-i}}) = Y_i(T_i, \mathbf{T}'_{\mathcal{N}_i}, \mathbf{X}'_{\mathcal{N}_i}, \mathbf{T}'_{\mathcal{N}_{-i}}, \mathbf{X}'_{\mathcal{N}_{-i}}).$$

(2) Unconfoundedness assumption: $Y_i(t_i, v_i) \perp T_i, V_i | \mathbf{X}_i, \forall t_i \in \{0, 1\}$ and $v_i \in \mathcal{V}_i$, for $i = 1, \dots, n$.

Hence, the expected response of one unit under network inference can be identified as $\mathbb{E}[Y_i(t_i, v_i)] = \mathbb{E}[Y_i | T_i = t_i, V_i = v_i, \mathbf{X}_i], \forall t_i \in \{0, 1\}$, and $v_i \in \mathcal{V}_i$, for $i = 1, \dots, n$. It is derived by

$$\begin{aligned} \mathbb{E}[Y_i | T_i = t_i, V_i = v_i, \mathbf{X}_i] &\stackrel{A_{sm.}(1)}{=} \mathbb{E}[Y_i(t_i, v_i) | T_i = t_i, V_i = v_i, \mathbf{X}_i] \\ &\stackrel{A_{sm.}(2)}{=} \mathbb{E}[Y_i(t_i, v_i) | \mathbf{X}_i]. \end{aligned}$$

B Synthetic Randomized Experiments on Wave1

On the in-school friendship network derived from the Wave1 questionnaire data, we conduct randomized intervention experiments that simulate the improvement of performance after assigning a student to a tutoring or support program. Recall that $Y_i(T_i = 0, \mathcal{G} = \emptyset)$ indicates the overall performance of student i before assigning it to a tutoring program or being influenced by peers. We select specific questions from the questionnaire and regard the corresponding answers as the features of corresponding students. These feature vectors are further used to construct a symmetrized k -NN similarity graph as the in-school friendship network. Questions related to the potential performance of students are list in Table 7.

Using the answers of selected questions and their abbreviations, $Y_i(T_i = 0, \mathcal{G} = \emptyset)$ is generated as follows

$$\begin{aligned} Y_i(T_i = 0, \mathcal{G} = \emptyset) &:= -X_{i,H1GH52} + 2X_{i,H1ED3} - X_{i,H1ED5} - 2X_{i,H1ED7} \\ &\quad - 0.5(X_{i,H1ED11} + X_{i,H1ED12} + X_{i,H1ED13} + X_{i,H1ED14}) \\ &\quad + 0.5(X_{i,H1DA5} + X_{i,H1DA7}) - 3X_{i,H1DS12} + f_{\mathcal{N}}(X_{i,H1HS1} \\ &\quad + X_{i,H1HS3} + X_{i,H1WP17B} + X_{i,H1TO51} + X_{i,H1TO53} \\ &\quad + X_{i,H1NB5} + X_{i,H1EE3} + X_{i,PA57D}), \end{aligned}$$

where $f_{\mathcal{N}}(\cdot)$ represents a 1-layer neural network with random coefficients.

The generating process of the individual treatment response also depends on the selected properties. For example, by assigning a student who has repeated grade will probably improve this student's performance. The treatment effect is simulated as follows:

$$\begin{aligned} \tau(\mathbf{X}_i) &:= X_{i,H1ED3} + 0.5(X_{i,H1GH52} + X_{i,H1ED5} + X_{i,H1ED7}) \\ &\quad + 0.5(X_{i,H1ED11} + X_{i,H1ED12} + X_{i,H1ED13} + X_{i,H1ED14}) \\ &\quad + X_{i,H1DS12} + f_{\mathcal{N}}, \end{aligned}$$

Table 7: Selected questions from the Wave1 data [13] that are used as feature vectors.

H1GH52	Do you get enough sleep?
H1ED3	Have you skipped a grade?
H1ED5	Have you repeated a grade?
H1ED7	Have you received an suspension?
H1HS1	Have you had a routine physical examination?
H1HS3	Have you received psychological counseling?
H1WP17B	Played a sport in the past 4 weeks?
H1TO51	Is alcohol easily available in your home?
H1TO53	Is a gun easily available in your home?
H1NB5	Do you feel safe in your neighborhood?
H1EE3	Did you work for pay in the last 4 weeks?
PA57D	Food stamps?
H1DA5	How often do you play sport?
H1DA7	How do you hang out with friends?
H1ED11	Your grade in English or language arts?
H1ED12	Your grade in mathematics?
H1ED13	Your grade in history or social studies?
H1ED14	Your grade in science?
H1DS12	How often did you sell marijuana or other drugs

where $f_{\mathcal{N}}$ represents a nonlinear random function depending on the rest of variables. Furthermore, peer effect in this synthetic experiment is generated by

$$\delta_i(\mathbf{X}, \mathbf{T}, \mathcal{G}) := \alpha \frac{1}{|\mathcal{N}_i|} \sum_{j \in \mathcal{N}_i} T_j \tau(\mathbf{X}_j), \quad (8)$$

where the decay parameter α characterizes the decay of influence. Eq. 8 means that the peer effect applied to the node i is determined by individual treatment responses of its neighbors who are under treatment. Finally, the outcome, e.g., the linear response G_0 , is simulated by

$$Y_i = Y_i(T_i = 0, \mathcal{G} = \emptyset) + T_i \tau(\mathbf{X}_i) + \delta_i(\mathbf{X}, \mathbf{T}, \mathcal{G}) + \epsilon_{Y_i}. \quad (9)$$

Other nonlinear response generation functions studied in the main text are defined as

$$G_1 : Y_i = Y_i(T_i = 0, \mathcal{G} = \emptyset) + T_i \tau(\mathbf{X}_i) + \delta_i(\mathbf{X}, \mathbf{T}, \mathcal{G}) + \kappa \delta_i^2(\mathbf{X}, \mathbf{T}, \mathcal{G}) + \epsilon_{Y_i}, \quad (10)$$

and

$$G_2 : Y_i = Y_i(T_i = 0, \mathcal{G} = \emptyset) + T_i \tau(\mathbf{X}_i) + \delta_i(\mathbf{X}, \mathbf{T}, \mathcal{G}) + \frac{\kappa}{2} \delta_i^2(\mathbf{X}, \mathbf{T}, \mathcal{G}) + \frac{\kappa}{2} \tau(\mathbf{X}_i) \delta_i(\mathbf{X}, \mathbf{T}, \mathcal{G}) + \epsilon_{Y_i}, \quad (11)$$

where κ characterizes the strength of nonlinear effects.

The benefit of using synthetic data is that we can modify the experiment settings. Three parameters control the experimental settings: number of neighbors k , which determines the graph structure and density; the probability p of assigning a node to treatment which controls the population imbalance between treatment and control groups; the decay parameter α , which determines the intensity of peer effect. For the evaluation results reported in the **main text** we generate the simulation data with parameters $k = 10$, $p = 0.1$, and $\alpha = 0.5$. We report more evaluations in Table 8, Table 9, and Table 10. One observation is that in the randomized experiment setting with linear response, the GraphSAGE-based estimator is

Table 8: Evaluation metrics on under-treated synthetic data with $\mathbf{p} = 0.1$, $\alpha = 0.5$, and $k = 5, 10$. Improvements are obtained by comparing with the GPS baseline. Both representation balancing $H\hat{S}IC^\Phi$ and $H\hat{S}IC^{GNN}$ are deployed in the GNN-based estimators for searching for the best performance.

	$k = 5$		$k = 10$	
	\sqrt{MSE}	ϵ_{PEHE}	\sqrt{MSE}	ϵ_{PEHE}
GPS	0.279 ± 0.071	0.210 ± 0.043	0.281 ± 0.049	0.139 ± 0.052
GCN	0.212 ± 0.035	0.095 ± 0.055	0.211 ± 0.013	0.058 ± 0.036
GraphSAGE	0.200 ± 0.032	0.088 ± 0.054	0.199 ± 0.030	0.057 ± 0.039
1-GNN	0.214 ± 0.039	0.096 ± 0.062	0.203 ± 0.033	0.057 ± 0.040
Improve	28.3%	58.1%	29.2%	59.0%

Table 9: Evaluation metrics on over-treated synthetic data with $\mathbf{p} = 0.7$, $\alpha = 0.5$, and $k = 5, 10$. Improvements are obtained by comparing with the GPS baseline. Both representation balancing $H\hat{S}IC^\Phi$ and $H\hat{S}IC^{GNN}$ are deployed in the GNN-based estimators for searching for the best performance.

	$k = 5$		$k = 10$	
	\sqrt{MSE}	ϵ_{PEHE}	\sqrt{MSE}	ϵ_{PEHE}
GPS	0.318 ± 0.010	0.409 ± 0.008	0.363 ± 0.087	0.491 ± 0.200
GCN	0.277 ± 0.007	0.051 ± 0.007	0.288 ± 0.063	0.087 ± 0.053
GraphSAGE	0.276 ± 0.024	0.050 ± 0.007	0.301 ± 0.054	0.083 ± 0.033
1-GNN	0.249 ± 0.006	0.054 ± 0.015	0.278 ± 0.056	0.076 ± 0.034
Improve	21.7%	87.8%	23.4%	84.5%

Table 10: Evaluation metrics on balanced synthetic data with $\mathbf{p} = 0.5$, $\alpha = 0.5$, and $k = 5, 10$. Improvements are obtained by comparing with the GPS baseline. Both representation balancing $H\hat{S}IC^\Phi$ and $H\hat{S}IC^{GNN}$ are deployed in the GNN-based estimators for searching for the best performance.

	$k = 5$		$k = 10$	
	\sqrt{MSE}	ϵ_{PEHE}	\sqrt{MSE}	ϵ_{PEHE}
GPS	0.329 ± 0.005	0.207 ± 0.015	0.294 ± 0.008	0.224 ± 0.071
GCN	0.269 ± 0.011	0.047 ± 0.006	0.215 ± 0.020	0.050 ± 0.012
GraphSAGE	0.279 ± 0.015	0.044 ± 0.003	0.223 ± 0.018	0.037 ± 0.011
1-GNN	0.268 ± 0.015	0.042 ± 0.005	0.214 ± 0.015	0.032 ± 0.007
Improve	18.5%	79.7%	27.2%	85.7%

a good candidate for causal inference in an under-treated population, while 1-GNN-based estimator is superior in a balanced- or over-treated population.

Table 11 reports the performance of GNN-based causal estimators on the Wave1 dataset using nonlinear response models. Nonlinear responses are generated via G_1 and G_2 under $\kappa = 0.5$. For the \sqrt{MSE} metric, GNN-based estimators outperform the best baseline by 23.6%(G_1) and 20.1%(G_2) on Wave1. Moreover, GNN-based causal estimators significantly outperform the best baseline in the individual treatment effect recovery task, where a 73.2%(G_1) and a 70.5%(G_2) improvement are observed.

Table 11: Experimental results of randomized experiments on the Wave1 dataset using nonlinear response generation functions G_1 and G_2 with $\kappa = 0.5$. Other parameters are set as (node degree) $k = 10$, (decay parameter) $\alpha = 0.5$, and (treatment probability) $p = 0.1$. Both representation balancing $H\hat{S}IC^\Phi$ and $H\hat{S}IC^{GNN}$ are deployed in the GNN-based estimators for searching for the best performance. Improvements are obtained by comparing with the best baseline.

	G_1		G_2	
	\sqrt{MSE}	ϵ_{PEHE}	\sqrt{MSE}	ϵ_{PEHE}
DA GB	0.742 ± 0.083	0.210 ± 0.008	1.060 ± 0.047	0.400 ± 0.054
DA RF	1.007 ± 0.027	0.527 ± 0.141	1.243 ± 0.089	1.056 ± 0.222
DR GB	0.784 ± 0.019	0.352 ± 0.074	1.116 ± 0.106	0.633 ± 0.195
DR EN	0.882 ± 0.053	0.575 ± 0.015	1.258 ± 0.176	0.841 ± 0.293
GPS	0.280 ± 0.017	0.142 ± 0.032	0.289 ± 0.012	0.244 ± 0.066
GCN + $H\hat{S}IC^\Phi / GNN$	0.224 ± 0.008	0.038 ± 0.003	0.237 ± 0.020	0.095 ± 0.010
GraphSAGE + $H\hat{S}IC^\Phi / GNN$	0.214 ± 0.007	0.045 ± 0.002	0.231 ± 0.014	0.072 ± 0.003
1-GNN + $H\hat{S}IC^\Phi / GNN$	0.216 ± 0.003	0.040 ± 0.001	0.250 ± 0.020	0.103 ± 0.015
Improve	23.6%	73.2%	20.1%	70.5%

C Synthetic Randomized Experiments on Pokec

The motivation for using a real social network dataset is that the k -NN similarity graph can violate the power-law degree distribution, as shown in Fig. 2. Consider hypothetical intervention experiments to the users of the Pokec social network. After reading a personalized advertisement or getting influenced by social contacts, a user is encouraged to purchase a new medicine. To simulate the individual buying behavior, we use profile features that are related to the health condition of a user. Table 12 lists the related features used in semi-synthetic experiments.

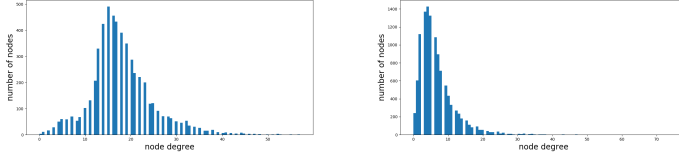


Fig. 2: Number of nodes vs. node degree from the k -NN similarity graph of Wave1 with $k = 10$ (left), and from the Pokec social network (right).

We assume that a healthy person with good habits is self-motivated to purchase health medicine even without external influences. Hence, $Y_i(T_i = 0, \mathcal{G} = \emptyset)$ is simulated as follows:

$$Y_i(T_i = 0, \mathcal{G} = \emptyset) := 0.2(1 - X_{i,gender}) + 0.5X_{i,age} - 0.2X_{i,weight} + 0.5X_{i,education} - 0.6(3 - X_{i,smoke}) + 0.2X_{i,sex} - 0.6(3 - X_{i,alcohol}) + \epsilon,$$

where ϵ is a Gaussian random variable with mean 0.1. Suppose that new health medicine is advertised to offer miraculous effects on weight loss, quit smoking, abstinence, etc. Then the individual treatment response can be generated by

$$\tau(\mathbf{X}_i) := 0.8(1 - X_{i,gender}) + X_{i,age} + 0.3X_{i,weight} + 0.5(1 - X_{i,eyesight}) + 0.5(X_{i,education} + 0.5) + 0.6X_{i,smoke} + 0.5X_{i,alcohol} + \epsilon.$$

Table 12: Characteristics of users and corresponding ranges of values selected from the Pokec social network data.

features	values
gender	[0, 1]
age	[15, 16, ..., 60]
height	[140, 141, ..., 200]
weight	[30, 31, ..., 200]
completed level of education	[0, 1, 2, 3]
eyesight	[0, 1]
relation to smoking	[0, 1, 2, 3]
relation to alcohol	[0, 1, 2, 3]
relation to casual sex	[0, 1, 2]

Since Pokec is a social network, in the semi-synthetic experiments, we also take into account long-range influences to simulate opinion propagation in the social network. To be more specific, the spillover effect on one node not only depends on the nearest neighboring nodes but also next-nearest neighboring nodes. Formally, it is defined as

$$\delta_i(\mathbf{X}, \mathbf{T}, \mathcal{G}) := \alpha \frac{1}{|\mathcal{N}_i|} \sum_{j \in \mathcal{N}_i} T_j \tau(\mathbf{X}_j) + \alpha^2 \frac{1}{|\mathcal{N}_i^{(2)}|} \sum_{k \in \mathcal{N}_i^{(2)}} T_k \tau(\mathbf{X}_k), \quad (12)$$

where α is the decay factor and $\mathcal{N}_i^{(2)}$ represents the next-nearest neighbors of i . Finally, the observed data in the randomized experiments can be derived from $Y_i(T_i = 0, \mathcal{G} = \emptyset), \tau(\mathbf{X}_i)$, and social network structure \mathcal{G}_{Pokec} using Eq. 12 and Eq. 9 for the linear response or Eq. 10 and Eq. 11 for nonlinear responses. The experiments reported in the **main text** use the setting $\alpha = 0.5$ and $p = 0.1$.

Table 13: Evaluation metrics on under-treated Pokec social network with $\mathbf{p} = \mathbf{0.1}$, $\alpha = 0.1, 0.9$. Improvements are obtained by comparing with the GPS baseline. Both representation balancing $H\hat{S}IC^\Phi$ and $H\hat{S}IC^{GNN}$ are deployed in the GNN-based estimators for searching for the best performance.

	$\alpha = 0.1$		$\alpha = 0.9$	
	\sqrt{MSE}	ϵ_{PEHE}	\sqrt{MSE}	ϵ_{PEHE}
GPS	0.263 ± 0.001	0.156 ± 0.017	0.595 ± 0.005	0.185 ± 0.005
GCN	0.230 ± 0.017	0.147 ± 0.031	0.573 ± 0.033	0.163 ± 0.005
GraphSAGE	0.227 ± 0.005	0.128 ± 0.015	0.569 ± 0.032	0.151 ± 0.011
1-GNN	0.231 ± 0.006	0.132 ± 0.014	0.571 ± 0.033	0.197 ± 0.020
Improve	13.5%	17.9%	4.4%	18.4%

Since the network structure \mathcal{G}_{Pokec} is given, we provide more experiment results in Table 13 and Table 14 to understand the effect of decay parameter α . In particular, we consider regimes from negligible peer effects with $\alpha = 0.1$ to significant peer effects with $\alpha = 0.9$. Since the covariates of neighboring units in the Pokec dataset have strong cosine similarity, and the simulation generation process is relatively simple, GNN-based causal estimators might overfit the superimposed causal effects and poorly recover the individual treatment effect. It is becoming more evident if the peer effects are strong and the population is over-treated, where the GPS baseline can achieve comparable results as other GNN-based estimators using only the information of exposure level (see Table 14).

Table 14: Evaluation metrics on over-treated Pokec social network with $p = 0.7$, $\alpha = 0.1, 0.9$. Improvements are obtained by comparing with the GPS baseline. Both representation balancing $H\hat{S}IC^\Phi$ and $H\hat{S}IC^{GNN}$ are deployed in the GNN-based estimators for searching for the best performance.

	$\alpha = 0.1$		$\alpha = 0.9$	
	\sqrt{MSE}	ϵ_{PEHE}	\sqrt{MSE}	ϵ_{PEHE}
GPS	0.404 ± 0.007	0.126 ± 0.004	1.438 ± 0.000	0.533 ± 0.015
GCN	0.247 ± 0.008	0.044 ± 0.003	1.426 ± 0.030	0.594 ± 0.039
GraphSAGE	0.240 ± 0.006	0.041 ± 0.001	1.417 ± 0.021	0.662 ± 0.061
1-GNN	0.233 ± 0.001	0.039 ± 0.002	1.390 ± 0.033	1.076 ± 0.094
Improve	42.3%	69.0%	3.3%	-11.4%

Table 15: Experimental results of randomized experiments on the Pokec dataset using nonlinear response generation functions G_1 and G_2 with $\kappa = 0.5$. We set the treatment probability as $p = 0.1$ and the decay parameter as $\alpha = 0.5$. Both representation balancing $H\hat{S}IC^\Phi$ and $H\hat{S}IC^{GNN}$ are deployed in the GNN-based estimators for searching for the best performance. Improvements are obtained by comparing with the best baselines.

	G_1		G_2	
	\sqrt{MSE}	ϵ_{PEHE}	\sqrt{MSE}	ϵ_{PEHE}
DA GB	1.342 ± 0.070	0.551 ± 0.026	2.095 ± 0.070	0.828 ± 0.282
DA RF	1.369 ± 0.060	1.015 ± 0.074	2.125 ± 0.080	1.389 ± 0.109
DR GB	1.324 ± 0.081	0.306 ± 0.011	2.038 ± 0.090	0.438 ± 0.005
DR EN	1.325 ± 0.078	0.336 ± 0.032	2.043 ± 0.089	0.338 ± 0.040
GPS	0.693 ± 0.058	0.450 ± 0.042	0.813 ± 0.068	0.375 ± 0.089
GCN + $H\hat{S}IC^{\Phi/GNN}$	0.483 ± 0.010	0.193 ± 0.001	0.729 ± 0.007	0.242 ± 0.032
GraphSAGE + $H\hat{S}IC^{\Phi/GNN}$	0.480 ± 0.009	0.198 ± 0.004	0.713 ± 0.017	0.217 ± 0.025
1-GNN + $H\hat{S}IC^{\Phi/GNN}$	0.454 ± 0.003	0.159 ± 0.005	0.767 ± 0.023	0.218 ± 0.002
Improve	34.5%	48.0%	12.3%	35.8%

Table 15 reports the performance of GNN-based causal estimators on the Pokec dataset using nonlinear response models. Nonlinear responses are generated via G_1 and G_2 under $\kappa = 0.5$. For the \sqrt{MSE} metric, GNN-based estimators outperform the best baseline by 34.5%(G_1) and 12.3%(G_2) on Wave1. Moreover, GNN-based causal estimators significantly outperform the best baseline in the individual treatment effect recovery task, where a 48.0%(G_1) and a 35.8%(G_2) improvement are observed.

D Additional Experiments for Intervention Policy Optimization

In addition to the policy optimization experiments on the Wave1 and Pokec simulation data under the treatment capacity constraint $p_t = 0.3$, in Table 16 we also report the intervention policy improvement under the treatment capacity constraint with $p_t = 0.5$.

Until now, we have only employed a simple neural network as the policy network with feature vectors as input. For GNN-based methods, the policy learner can adjust its treatment rules according to the neighboring nodes' features and responses through the GNN-based causal estimators. However, through baseline estimators, e.g., doubly-robust estimators, a simple policy network cannot access the neighboring features of a node. Therefore, for a fair comparison, we employ another 1-GNN as the policy network, and the evaluations on the

Table 16: Intervention policy improvements on the Wave1 and Pokec semi-synthetic datasets under treatment capacity constraint with $p_t = 0.5$. Note that *only* $\Delta S(\hat{\pi}_n^{p_t})$ reflects the genuine policy improvement.

	Wave1		Pokec	
	$\Delta \hat{S}(\hat{\pi}_n^{p_t})$	$\Delta S(\hat{\pi}_n^{p_t})$	$\Delta \hat{S}(\hat{\pi}_n^{p_t})$	$\Delta S(\hat{\pi}_n^{p_t})$
DA GB	0.636 ± 0.028	0.012 ± 0.025	0.479 ± 0.066	0.002 ± 0.055
DA RF	0.644 ± 0.027	0.016 ± 0.023	0.477 ± 0.049	0.008 ± 0.045
DR GB	0.761 ± 0.037	0.003 ± 0.031	0.712 ± 0.133	0.001 ± 0.089
DR EN	0.901 ± 0.150	0.006 ± 0.100	0.708 ± 0.093	0.001 ± 0.078
GPS	0.964 ± 0.091	0.018 ± 0.076	0.841 ± 0.072	0.007 ± 0.060
GCN	0.725 ± 0.015	0.544 ± 0.012	0.747 ± 0.041	0.566 ± 0.035
GraphSAGE	0.712 ± 0.031	0.532 ± 0.024	0.754 ± 0.099	0.559 ± 0.079
1-GNN	0.722 ± 0.052	0.546 ± 0.041	0.806 ± 0.031	0.586 ± 0.023

Table 17: Intervention policy improvements on the Wave1 semi-synthetic dataset under treatment capacity constraint with $p_t = 0.3$. The policy network employed is another 1-GNN. Note that *only* $\Delta S(\hat{\pi}_n^{p_t})$ reflects the real policy improvement.

	Wave1	
	$\Delta \hat{S}(\hat{\pi}_n^{p_t})$	$\Delta S(\hat{\pi}_n^{p_t})$
DA GB	0.291 ± 0.031	0.004 ± 0.026
DA RF	0.310 ± 0.041	0.003 ± 0.032
DR GB	0.102 ± 0.057	0.002 ± 0.048
DR EN	0.360 ± 0.044	0.002 ± 0.037
GPS	0.278 ± 0.061	0.006 ± 0.051
GCN	0.279 ± 0.029	0.179 ± 0.026
GraphSAGE	0.268 ± 0.023	0.169 ± 0.019
1-GNN	0.310 ± 0.022	0.201 ± 0.016

Wave1 dataset are given in Table 17. The results further confirm that the accuracy of causal effect estimators is crucial for intervention policy optimization on interconnected units.

E Experiment Settings

E.1 GNN-based Estimators in Causal Inference Experiments

For GNN-based estimators, we use Adam as a default optimizer with learning rate 0.001 and weight decay 0.0001. The number of total epochs is 20,000; early stopping is employed by monitoring the loss on the validation set every 2000 epochs. Hyperparameter κ in \mathcal{L}_{est} for penalizing the distribution discrepancy is searched from $\{0.001, 0.005, 0.1, 0.2\}$ for the Wave1 and Pokec datasets, and from $\{0.1, 0.2, 0.5, 1.\}$ for the Amazon dataset. The feature map neural network Φ has hidden dimensions $[64, 64]$ for the Wave1 and Pokec datasets, and $[256, 128, 128]$ for the Amazon dataset. GNNs have hidden dimensions $[128, 32]$ for the Wave1 and Pokec datasets, and $[256, 128, 64]$ for the Amazon dataset. Outcome prediction networks h_0 and h_1 have hidden dimensions $[64, 32]$ for the Wave1 and Pokec datasets, and $[256, 128, 64]$ for the Amazon dataset. ReLU is used as the activation function between hidden layers. Dropout is also employed between hidden layers with dropout rate a 0.5.

E.2 Baseline Estimators in Causal Inference Experiments

For baseline models, learning rate of the DR EN model is searched from $\{0.001, 0.01, 0.1\}$ with maximal iteration 10000. For the DA RF model, the number of estimators is searched from $\{5, 10, 20\}$, the maximal depth from $\{5, 10, 20\}$, and the minimum number of samples at a leaf node from $\{5, 10, 20\}$. For the DR GB and DA GB models, the number of estimators is searched from $\{10, 50\}$, and the maximal depth is searched from $\{5, 10\}$. In our experiments, the training procedure of Domain Adaption estimators for causal inference under interference is given as below

$$\begin{aligned}\hat{\mu}_0 &= M_1 \left(Y_i^0 \sim [\mathbf{X}_i^0; G_i], \text{weights} = \frac{g(\mathbf{X}_i^0)}{1 - g(\mathbf{X}_i^0)} \right), \\ \hat{\mu}_1 &= M_2 \left(Y_i^1 \sim [\mathbf{X}_i^1; G_i], \text{weights} = \frac{1 - g(\mathbf{X}_i^1)}{g(\mathbf{X}_i^1)} \right), \\ \hat{D}_i^1 &= Y_i^1 - \hat{\mu}_0([\mathbf{X}_i^1; G_i]), \\ \hat{D}_i^0 &= \hat{\mu}_1([\mathbf{X}_i^0; G_i]) - Y_i^0, \\ \hat{\tau} &= M_3(\hat{D}_i^0 | \hat{D}_i^1 \sim \mathbf{X}_i^0 | \mathbf{X}_i^1),\end{aligned}$$

where M_1, M_2, M_3 are machine learning algorithms; Y_i^0, \mathbf{X}_i^0 represent the outputs and covariates of units under control in the training dataset, and Y_i^1, \mathbf{X}_i^1 under treatment. To capture the interference, the exposure variable G_i is concatenated to the covariates. $g(\mathbf{X}_i)$ is an estimation of $\Pr[T_i = 1 | \mathbf{X}_i]$ in the observational study using the Amazon dataset, while it is the predefined treatment probability p in randomized experiments using the Wave1 and Pokec datasets. Similarly, the training procedure of Doubly Robust estimators for causal inference under interference is given as

$$\begin{aligned}\hat{\mu}_0 &= M_1(Y_i^0 \sim [\mathbf{X}_i^0; G_i]), \\ \hat{\mu}_1 &= M_2(Y_i^1 \sim [\mathbf{X}_i^1; G_i]), \\ \hat{D}_i^1 &= \hat{\mu}_1([\mathbf{X}_i; G_i]) + \frac{Y_i - \hat{\mu}_1([\mathbf{X}_i; G_i])}{g(\mathbf{X}_i)} \mathbb{1}_{\{T_i = 1\}}, \\ \hat{D}_i^0 &= \hat{\mu}_0([\mathbf{X}_i; G_i]) + \frac{Y_i - \hat{\mu}_0([\mathbf{X}_i; G_i])}{1 - g(\mathbf{X}_i)} \mathbb{1}_{\{T_i = 0\}}, \\ \hat{\tau} &= M_3((\hat{D}_i^1 - \hat{D}_i^0) \sim \mathbf{X}_i),\end{aligned}$$

where M_1, M_2, M_3 are machine learning algorithms; $g(\mathbf{X}_i)$ is an estimation of $\Pr[T_i = 1 | \mathbf{X}_i]$ in the observational study using the Amazon dataset, while it is the predefined treatment probability p in randomized experiments using the Wave1 and Pokec datasets.

E.3 Intervention Policy Experiments

Causal estimators with the best performance will be saved and fixed for the subsequent intervention policy improvement experiments on the same dataset. We use Adam as a default optimizer for the policy network with a learning rate of 0.001. The policy network has hidden dimensions $[64, 32]$ for the Wave1 and Pokec datasets, and $[128, 64, 64]$ for the Amazon dataset. ReLU is employed as the activation function between hidden layers, and a sigmoid function is applied to the output. Treatment is then sampled from a Bernoulli distribution

using the output of the policy network as the probability. The Gumbel-softmax trick [16] is employed such that errors can be back-propagated. Hyperparameter γ in \mathcal{L}_{pol} for enforcing the constraint is chosen from $\{5, 50, 100, 200, 500\}$, such that the pre-defined constraint can be satisfied within the tolerance ± 0.01 . Besides, we also penalize the distribution discrepancy under the new intervention policy given by the policy network, and the hyperparameter for penalizing this term is chosen from $\{0.0, 0.0001, 0.001, 0.01, 0.1, 1\}$. The number of training epochs is 2000, and each experiment is repeated 5 times.

F Omitted Proofs

F.1 Omitted Proof of Theorem 1

Recall that throughout the estimation of intervention policy regret bound, we keep the following assumptions.

Assumption 3.

(BO) *Bounded treatment and spillover effects:* There exist $0 < M_1, M_2 < \infty$ such that the individual treatment effect satisfies $|\tau_i| \leq M_1$ and the spillover effect satisfies $\forall \pi \in \Pi, |\delta_i(\pi)| \leq M_2$.

(WI) *Weak independence assumption:* For any node indices i and j , the weak independence assumption assumes that $\mathbf{X}_i \perp \mathbf{X}_j$ if $A_{ij} = 0$, or $\nexists k$ with $A_{ik} = A_{kj} = 1$.

(LIP) *Lipschitz continuity of the spillover effect w.r.t. policy:* Given two treatment policies π_1 and π_2 , for any node i the spillover effect satisfies $|\delta_i(\pi_1) - \delta_i(\pi_2)| \leq L \|\pi_1 - \pi_2\|_\infty$, where the Lipschitz constant satisfies $L > 0$ and $\|\pi_1 - \pi_2\|_\infty := \sup_{\mathbf{X} \in \mathcal{X}} |\pi_1(\mathbf{X}) - \pi_2(\mathbf{X})|$.

(ES) *Uniformly consistency:* after fitting experimental or observational data on \mathcal{G} , individual treatment effect estimator satisfies

$$\frac{1}{n} \sum_{i=1}^n |\tau_i - \hat{\tau}_i| < \frac{\alpha_\tau}{n\zeta_\tau},$$

and spillover estimator satisfies

$$\forall \pi \in \Pi, \frac{1}{n} \sum_{i=1}^n |\delta_i(\pi) - \hat{\delta}_i(\pi)| < \frac{\alpha_\delta}{n\zeta_\delta} \quad (13)$$

where $\alpha_\tau > 0$ and $\alpha_\delta > 0$ are scaling factors that characterize the errors of estimators. ζ_τ and ζ_δ control the convergence rate of estimators for individual treatment effect and spillover effect, respectively, which satisfy $0 < \zeta_\tau, \zeta_\delta < 1$.

The underlying difficulty of estimating the intervention policy regret is the networked setting. Weak independence assumption (WI) allows us to use hypergraph-based method and derive concentration inequalities for the networked random variables. This becomes a plausible assumption if the spillover effect only depends on the nearest neighbors and/or next-nearest neighbors. Note that the assumption (LIP) is plausible, at least, in the synthetic experiments. For instance, consider the spillover effect in the simulated experiments generated by $\delta_i(\pi) = \alpha \frac{1}{|\mathcal{N}_i|} \sum_{j \in \mathcal{N}_i} \pi(\mathbf{X}_j) \tau(\mathbf{X}_j)$ (see Eq. 8), then we can see

$$|\delta_i(\pi_1) - \delta_i(\pi_2)| \leq \alpha \frac{1}{|\mathcal{N}_i|} \sum_{j \in \mathcal{N}_i} M_1 |\pi_1(\mathbf{X}_j) - \pi_2(\mathbf{X}_j)| \leq \alpha M_1 \|\pi_1 - \pi_2\|_\infty.$$

Hence, in this example $L = \alpha M_1$.

Concentration inequalities on partly dependent random variables are first given in [17]. Later, [44] provides tighter concentration inequalities using hypergraph and weak dependence assumption. A hypergraph is a generalization of graph in which a hyperedge groups a number of vertices in the graph. For instance, consider a graph with n vertices, and let $\mathcal{N} = \{v_1, v_2, \dots, v_n\}$ represent the set of vertices. Hyperedges set $\mathcal{E}_h = \{e_{h,1}, e_{h,2}, \dots, e_{h,m}\}$ represents instances joining a number of vertices. In the following, let $\mathcal{G}_h = (\mathcal{N}, \mathcal{E}_h)$ denote a hypergraph.

Definition 1 (Definition 1 in [44]). *Given a hypergraph \mathcal{G}_h , we call $\{\xi_i\}_{i=1}^n$ \mathcal{G}_h -networked random variables if there exist functions $f_i : \mathcal{X}^{\otimes |e_{h,i}|} \rightarrow \mathbb{R}$ such that $\xi_i = f_i(\{\mathbf{X}_v | v \in e_{h,i}\})$, where $\{\mathbf{X}_v | v \in e_{h,i}\}$ represents the set of covariates of the vertices in the hyperedge $e_{h,i}$.*

Furthermore, we have the following concentration inequality.

Theorem 3 (Corollary 7 in [44]). *Let $\{\xi_i\}_{i=1}^n$ be \mathcal{G}_h -networked random variables with mean $\mathbb{E}[\xi_i] = \mu$, and satisfying $a < \xi_i < b, \forall i \in \{1, 2, \dots, n\}$. Then for all $\epsilon > 0$,*

$$\Pr \left(\left| \frac{1}{n} \sum_{i=1}^n \xi_i - \mu \right| \geq \epsilon \right) \leq \exp \left(-\frac{n\epsilon^2}{2\omega_{\mathcal{G}_h}(b-a)^2} \right), \quad (14)$$

where $\omega_{\mathcal{G}_h} := \max_{v \in \mathcal{N}} |\{e_h : v \in e_h\}|$ represents the maximal degree of \mathcal{G}_h .

Recall the following definitions of utility functions $S_n^{\tau, \delta}(\pi)$, $\hat{S}_n^{\tau, \delta}(\pi)$, and $S(\pi)$

$$\begin{aligned} S(\pi) &:= \mathbb{E}[(2\pi(\mathbf{X}_i) - 1)(\tau_i + \delta_i(\pi))] \\ S_n^{\tau, \delta}(\pi) &:= \frac{1}{n} \sum_{i=1}^n (2\pi(\mathbf{X}_i) - 1)(\tau_i + \delta_i(\pi)) \\ \hat{S}_n^{\tau, \delta}(\pi) &:= \frac{1}{n} \sum_{i=1}^n (2\pi(\mathbf{X}_i) - 1)(\hat{\tau}_i + \hat{\delta}_i(\pi)), \end{aligned}$$

where the policy π function has output in $[0, 1]$. An optimal empirical policy is obtained via $\hat{\pi}_n \in \arg\max_{\pi \in \Pi} \hat{S}_n^{\tau, \delta}(\pi)$. Note that in the definition of $S(\pi)$ we still keep the subindex i to emphasize the dependence of spillover effect on neighboring nodes. Next we provide several lemmas related to the utility functions.

Lemma 1. *Let $\mathcal{S}(\pi) := S_n^{\tau, \delta}(\pi) - S(\pi)$, for any $\pi_1, \pi_2 \in \Pi$, where the policy class is contained in $[0, 1]$, according to the assumptions (BO) and (LIP) we have*

$$|\mathcal{S}(\pi_1) - \mathcal{S}(\pi_2)| \leq 2(2M_1 + 2M_2 + L)\|\pi_1 - \pi_2\|_\infty$$

Proof. First note that $|\mathcal{S}(\pi_1) - \mathcal{S}(\pi_2)| \leq |S(\pi_1) - S(\pi_2)| + |S_n^{\tau, \delta}(\pi_1) - S_n^{\tau, \delta}(\pi_2)|$, and we have

$$\begin{aligned} |S(\pi_1) - S(\pi_2)| &= \left| \int_{\mathcal{X}} (2\pi_1(\mathbf{X}_i) - 1)(\tau_i + \delta_i(\pi_1)) - (2\pi_2(\mathbf{X}_i) - 1)(\tau_i + \delta_i(\pi_2)) d\mathbf{X}_i \right| \\ &\leq \int_{\mathcal{X}} 2|\tau_i| \|\pi_1 - \pi_2\|_\infty + |(2\pi_1(\mathbf{X}_i) - 1)(\delta_i(\pi_2) + L\|\pi_1 - \pi_2\|_\infty) - (2\pi_2(\mathbf{X}_i) - 1)\delta_i(\pi_2)| d\mathbf{X}_i \\ &= \int_{\mathcal{X}} 2|\tau_i| \|\pi_1 - \pi_2\|_\infty + |2(\pi_1(\mathbf{X}_i) - \pi_2(\mathbf{X}_i))\delta_i(\pi_2) + L(2\pi_1(\mathbf{X}_i) - 1)\|\pi_1 - \pi_2\|_\infty| d\mathbf{X}_i \\ &\leq (2|\tau_i| + 2|\delta_i(\pi_2)| + L)\|\pi_1 - \pi_2\|_\infty \\ &\leq (2M_1 + 2M_2 + L)\|\pi_1 - \pi_2\|_\infty. \end{aligned}$$

Similarly, we have $|S_n^{\tau, \delta}(\pi_1) - S_n^{\tau, \delta}(\pi_2)| \leq (2M_1 + 2M_2 + L)\|\pi_1 - \pi_2\|_\infty$. ■

Using the concentration inequality in Theorem 3 we can obtain the convergence rate of the worst-case utility regret. We also use a capacity measure of the policy functional class Π , namely the covering number, to prove the convergence rate, which is defined in the following.

Definition 2 (Definition 3.1 in [7]). *Let Π be a metric space and $\epsilon > 0$, the covering number $\mathcal{N}(\Pi, \epsilon)$ is defined as the minimal $l \in \mathbb{N}$ such that there exist l disks in Π with radius ϵ covering Π .*

Lemma 2. *Under Assumption 3, for any $\{\mathbf{X}_i\}_{i=1}^n \in \mathcal{X}^{\otimes n}$ and $\epsilon > 0$, it satisfies*

$$\Pr \left(\sup_{\pi \in \Pi} |S_n^{\tau, \delta}(\pi) - S(\pi)| \leq \epsilon \right) \geq 1 - \mathcal{N} \left(\Pi, \frac{\epsilon}{4(2M_1 + 2M_2 + L)} \right) \exp \left(-\frac{n\epsilon^2}{32(d_{\max}^2 + 1)(M_1 + M_2)^2} \right), \quad (15)$$

where $\mathcal{N} \left(\Pi, \frac{\epsilon}{4(2M_1 + 2M_2 + L)} \right)$ represents the covering number on the policy functional class Π with radius $\frac{\epsilon}{4(2M_1 + 2M_2 + L)}$.

Proof. According to the assumption (BO), the summands are bounded as $|(2\pi(\mathbf{X}_i) - 1)(\tau_i + \delta_i(\pi))| \leq M_1 + M_2, \forall i \in \{1, \dots, n\}$. Given the graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ and its corresponding adjacency matrix A , using the weak independence assumption (WI) a dependence hypergraph can be defined as $\mathcal{G}_h = (\mathcal{N}, \mathcal{E}_h)$, where a hyperedge $e_{h,i} \in \mathcal{E}_h$ is defined as $e_{h,i} := \{v_i\} \cup \{v_j | j \in \mathcal{N}_i\} \cup \{v_k | \exists j : A_{ij} = 1 \wedge A_{jk} = 1\}$. Therefore, the maximal degree of the hypergraph \mathcal{G}_h satisfies $\omega_{\mathcal{G}_h} \leq d_{\max}^2 + 1$, where d_{\max} indicates the maximal vertex degree of the graph \mathcal{G} . Via Theorem 3, we have

$$\Pr \left(|S_n^{\tau, \delta}(\pi) - S(\pi)| \geq \epsilon \right) \leq \exp \left(-\frac{n\epsilon^2}{8(d_{\max}^2 + 1)(M_1 + M_2)^2} \right), \quad \forall \pi \in \Pi. \quad (16)$$

Let $l = \mathcal{N} \left(\Pi, \frac{\epsilon}{2(2M_1 + 2M_2 + L)} \right)$ denote the covering number. Consider policies π_j , with $j \in \{1, \dots, l\}$ located in the center of disks D_j with radius $\frac{\epsilon}{2(2M_1 + 2M_2 + L)}$ which cover the policy functional class Π . Recall the definition $\mathcal{S}(\pi) := S_n^{\tau, \delta}(\pi) - S(\pi)$, by Lemma 1, for any π_j and $\pi \in D_j$, we have

$$|\mathcal{S}(\pi) - \mathcal{S}(\pi_j)| \leq 2(2M_1 + 2M_2 + L) \frac{\epsilon}{2(2M_1 + 2M_2 + L)} = \epsilon.$$

Then $\forall \pi \in D_j, \sup_{\pi \in D_j} \mathcal{S}(\pi) \geq 2\epsilon \Rightarrow \mathcal{S}(\pi_j) \geq \epsilon$, which indicates

$$\Pr \left(\sup_{\pi \in D_j} \mathcal{S}(\pi) \geq 2\epsilon \right) \leq \Pr(\mathcal{S}(\pi_j) \geq \epsilon) \leq \exp \left(-\frac{n\epsilon^2}{8(d_{\max}^2 + 1)(M_1 + M_2)^2} \right).$$

Since $\Pi = D_1 \cup \dots \cup D_l$, it is easy to see

$$\begin{aligned} \Pr \left(\sup_{\pi \in \Pi} \mathcal{S}(\pi) \geq 2\epsilon \right) &\leq \sum_{j=1}^l \Pr \left(\sup_{\pi \in D_j} \mathcal{S}(\pi) \geq 2\epsilon \right) \\ &\leq \mathcal{N} \left(\Pi, \frac{\epsilon}{2(2M_1 + 2M_2 + L)} \right) \exp \left(-\frac{n\epsilon^2}{8(d_{\max}^2 + 1)(M_1 + M_2)^2} \right). \end{aligned}$$

Upper bound for the probability $\Pr(\sup_{\pi \in \Pi} \mathcal{S}(\pi) \leq -2\epsilon)$ can be derived in the same way. The statement becomes valid by replacing ϵ by $\frac{\epsilon}{2}$. ■

Theorem 4 (Theorem 1 in the main text restated). *By Assumption 3, for any small $\epsilon > 0$, the policy regret is bounded by $\mathcal{R}(\hat{\pi}_n) \leq 2 \left(\frac{\alpha_\tau}{n\zeta_\tau} + \frac{\alpha_\delta}{n\zeta_\delta} \right) + 2\epsilon$ with probability at least $1 - \mathcal{N} \left(\Pi, \frac{\epsilon}{4(2M_1+2M_2+L)} \right) \exp \left(-\frac{n\epsilon^2}{32(d_{\max}^2+1)(M_1+M_2)^2} \right)$, where $\mathcal{N} \left(\Pi, \frac{\epsilon}{4(2M_1+2M_2+L)} \right)$ indicates the covering number on the functional class Π with radius $\frac{\epsilon}{4(2M_1+2M_2+L)}$, and d_{\max} is the maximal node degree in the graph \mathcal{G} .*

Proof. Consider an arbitrary policy $\tilde{\pi} \in \Pi$, we have the following utility difference

$$\begin{aligned} S(\tilde{\pi}) - S(\hat{\pi}_n) &= S_n^{\tau,\delta}(\tilde{\pi}) - S_n^{\tau,\delta}(\tilde{\pi}) + S_n^{\tau,\delta}(\hat{\pi}_n) - S_n^{\tau,\delta}(\hat{\pi}_n) \\ &\quad + S(\tilde{\pi}) - S(\hat{\pi}_n) + \hat{S}_n^{\tau,\delta}(\hat{\pi}_n) - \hat{S}_n^{\tau,\delta}(\hat{\pi}_n) \\ &\leq \underbrace{S_n^{\tau,\delta}(\tilde{\pi}) - \hat{S}_n^{\tau,\delta}(\tilde{\pi}) - S_n^{\tau,\delta}(\hat{\pi}_n) + \hat{S}_n^{\tau,\delta}(\hat{\pi}_n)}_{(1)} \\ &\quad + \underbrace{S(\tilde{\pi}) - S_n^{\tau,\delta}(\tilde{\pi}) + S_n^{\tau,\delta}(\hat{\pi}_n) - S(\hat{\pi}_n)}_{(2)}. \end{aligned}$$

Using $\forall \pi \in \Pi, \pi \in [0, 1]$ and assumption (ES) the term (\star) can be bounded as

$$\begin{aligned} (1) &= \frac{1}{n} \sum_{i=1}^n 2(\tau_i - \hat{\tau}_i)(\tilde{\pi}(\mathbf{X}_i) - \hat{\pi}_n(\mathbf{X}_i)) \\ &\quad + \frac{1}{n} \sum_{i=1}^n (2\tilde{\pi}(\mathbf{X}_i) - 1)(\delta_i(\tilde{\pi}) - \hat{\delta}_i(\tilde{\pi})) - \frac{1}{n} \sum_{i=1}^n (2\hat{\pi}_n(\mathbf{X}_i) - 1)(\delta_i(\hat{\pi}_n) - \hat{\delta}_i(\hat{\pi}_n)) \\ &\leq \frac{1}{n} \sum_{i=1}^n 2|\tau_i - \hat{\tau}_i| + \frac{1}{n} \sum_{i=1}^n |\delta_i(\tilde{\pi}) - \hat{\delta}_i(\tilde{\pi})| + \frac{1}{n} \sum_{i=1}^n |\delta_i(\hat{\pi}_n) - \hat{\delta}_i(\hat{\pi}_n)| \\ &\leq 2 \left(\frac{\alpha_\tau}{n\zeta_\tau} + \frac{\alpha_\delta}{n\zeta_\delta} \right). \end{aligned}$$

Furthermore, $(2) \leq |S_n^{\tau,\delta}(\tilde{\pi}) - S(\tilde{\pi})| + |S_n^{\tau,\delta}(\hat{\pi}_n) - S(\hat{\pi}_n)| \leq 2 \sup_{\pi \in \Pi} |S_n^{\tau,\delta}(\pi) - S(\pi)|$. In summary,

$$\mathcal{R}(\hat{\pi}_n) := \sup_{\tilde{\pi} \in \Pi} (S(\tilde{\pi}) - S(\hat{\pi}_n)) \leq 2 \left(\frac{\alpha_\tau}{n\zeta_\tau} + \frac{\alpha_\delta}{n\zeta_\delta} \right) + 2\epsilon,$$

with probability at least $1 - \mathcal{N} \left(\Pi, \frac{\epsilon}{4(2M_1+2M_2+L)} \right) \exp \left(-\frac{n\epsilon^2}{32(d_{\max}^2+1)(M_1+M_2)^2} \right)$ via Lemma 2. \blacksquare

F.2 Capacity-constrained Policy Regret

Before introducing a capacity-constrained utility function under network interference, we first review the definition of $A(\pi)$ following Section 2 of [3]. The benefit of deploying the intervention policy π compared to assigning everyone in control group is defined as

$$V(\pi) := \mathbb{E}[Y_i(T_i = 1)\pi(\mathbf{X}_i) + Y_i(T_i = 0)(1 - \pi(\mathbf{X}_i))] - \mathbb{E}[Y_i(T_i = 0)] = \mathbb{E}[\pi(\mathbf{X}_i)\tau(\mathbf{X}_i)],$$

and the utility function equals

$$A(\pi) := 2V(\pi) - \mathbb{E}[\tau(\mathbf{X}_i)] = \mathbb{E}[(2\pi(\mathbf{X}_i) - 1)\tau(\mathbf{X}_i)].$$

In the following, let us consider policy learning under treatment constraint p_t . If the distribution of covariates $\mathcal{P}_{\mathbf{X}}$ is known, and let $\mathcal{P}_{\mathbf{X}}(\pi)$ denote the treatment rule on the covariates space, then a capacity-constrained welfare gain relative to treating no one is defined as (see also Section 4.1 of [22])

$$\begin{aligned} V_{p_t}(\pi) &:= \mathbb{E}[[Y_i(T_i = 1) \min\{1, \frac{p_t}{\mathcal{P}_{\mathbf{X}}(\pi)}\} + Y_i(T_i = 0)(1 - \min\{1, \frac{p_t}{\mathcal{P}_{\mathbf{X}}(\pi)}\})]\pi(\mathbf{X}_i) \\ &\quad + Y_i(T_i = 0)(1 - \pi(\mathbf{X}_i))] - \mathbb{E}[Y_i(T_i = 0)] \\ &= \min\{1, \frac{p_t}{\mathcal{P}_{\mathbf{X}}(\pi)}\} \mathbb{E}[\pi(\mathbf{X}_i)\tau(\mathbf{X}_i)], \end{aligned}$$

and the corresponding capacity-constrained utility function equals

$$A_{p_t}(\pi) := 2V_{p_t}(\pi) - \mathbb{E}[\tau(\mathbf{X}_i)] = \mathbb{E}[(2 \min\{1, \frac{p_t}{\mathcal{P}_{\mathbf{X}}(\pi)}\} \pi(\mathbf{X}_i) - 1)\tau(\mathbf{X}_i)].$$

Similarly, the capacity-constrained utility function under interference for interconnected units reads

$$S_{p_t}(\pi) := \mathbb{E}[(2 \min\{1, \frac{p_t}{\mathcal{P}_{\mathbf{X}}(\pi)}\} \pi(\mathbf{X}_i) - 1)(\tau_i + \delta_i(\pi))].$$

Moreover, the empirical version of $S_{p_t}(\pi)$ reads

$$S_{n,p_t}^{\tau,\delta}(\pi) := \frac{1}{n} \sum_{i=1}^n (2 \min\{1, \frac{p_t}{\mathcal{P}_{\mathbf{X}}(\pi)}\} \pi(\mathbf{X}_i) - 1)(\tau_i + \delta_i(\pi)).$$

The empirical estimation of $S_{p_t}(\pi)$ with causal estimators being plugged in reads

$$\hat{S}_{n,p_t}^{\tau,\delta}(\pi) := \frac{1}{n} \sum_{i=1}^n (2 \min\{1, \frac{p_t}{\mathcal{P}_{\mathbf{X}}(\pi)}\} \pi(\mathbf{X}_i) - 1)(\hat{\tau}_i + \hat{\delta}_i(\pi)),$$

an corresponding optimal capacity-constrained policy is obtained via ⁵

$$\hat{\pi}_n^{p_t} \in \operatorname{argmax}_{\pi \in \Pi} \hat{S}_{n,p_t}^{\tau,\delta}(\pi).$$

Moreover, let $\pi^{p_t^*}$ denote the best possible intervention policy from the functional class Π with respect to the utility $S_{p_t}(\pi)$, namely $\pi^{p_t^*} \in \operatorname{argmax}_{\pi \in \Pi} S_{p_t}(\pi)$. The capacity-constrained policy regret is defined as $\mathcal{R}(\hat{\pi}_n^{p_t}) := S_{p_t}(\pi^{p_t^*}) - S_{p_t}(\hat{\pi}_n^{p_t})$. Before estimating the capacity-constrained intervention policy regret we derive the following inequality similar to Lemma 1.

Lemma 3. *Let $S_{p_t}(\pi) := S_{n,p_t}^{\tau,\delta}(\pi) - S_{p_t}(\pi)$, for any $\pi_1, \pi_2 \in \Pi$, where the policy class in contained in $[0, 1]$, according to the assumptions (BO) and (LIP) we have*

$$|S_{p_t}(\pi_1) - S_{p_t}(\pi_2)| \leq 4[(M_1 + M_2 + L) + \frac{1}{p_t}(M_1 + M_2)]\|\pi_1 - \pi_2\|_{\infty}. \quad (17)$$

⁵ This optimal capacity-constrained policy is, in principle, equivalent to the one obtained by minimizing the loss function $\mathcal{L}_{\text{pol}}(\pi) := -\hat{S}_n^{\tau,\delta}(\pi) + \gamma(\frac{1}{n} \sum_{i=1}^n \pi(\mathbf{X}_i) - p_t)$, since, in practice, treatment capacity constraint can be satisfied via Lagrangian multiplier.

Proof. Note that $|S_{p_t}(\pi_1) - S_{p_t}(\pi_2)| \leq |S_{p_t}(\pi_1) - S_{p_t}(\pi_2)| + |S_{n,p_t}^{\tau,\delta}(\pi_1) - S_{n,p_t}^{\tau,\delta}(\pi_2)|$. We first rewrite $S_{p_t}(\pi)$ as

$$S_{p_t}(\pi) = \min\{1, \frac{p_t}{\mathcal{P}_{\mathbf{X}}(\pi)}\} \mathbb{E}[(2\pi(\mathbf{X}_i) - 1)(\tau_i + \delta_i(\pi))] + (\min\{1, \frac{p_t}{\mathcal{P}_{\mathbf{X}}(\pi)}\} - 1) \mathbb{E}[\tau_i + \delta_i(\pi)].$$

Recall the definition of $S(\pi)$, and define $T(\pi) := \mathbb{E}[\tau_i + \delta_i(\pi)]$, we have

$$\begin{aligned} |S_{p_t}(\pi_1) - S_{p_t}(\pi_2)| &= |\min\{1, \frac{p_t}{\mathcal{P}_{\mathbf{X}}(\pi_1)}\} S(\pi_1) - \min\{1, \frac{p_t}{\mathcal{P}_{\mathbf{X}}(\pi_2)}\} S(\pi_2)| \\ &\quad + |(\min\{1, \frac{p_t}{\mathcal{P}_{\mathbf{X}}(\pi_1)}\} - 1)T(\pi_1) - (\min\{1, \frac{p_t}{\mathcal{P}_{\mathbf{X}}(\pi_2)}\} - 1)T(\pi_2)| \\ &\leq |\min\{1, \frac{p_t}{\mathcal{P}_{\mathbf{X}}(\pi_1)}\}| |S(\pi_1) - S(\pi_2)| \\ &\quad + |S(\pi_2)| |\min\{1, \frac{p_t}{\mathcal{P}_{\mathbf{X}}(\pi_1)}\} - \min\{1, \frac{p_t}{\mathcal{P}_{\mathbf{X}}(\pi_2)}\}| \\ &\quad + |\min\{1, \frac{p_t}{\mathcal{P}_{\mathbf{X}}(\pi_1)}\} - 1| |T(\pi_1) - T(\pi_2)| \\ &\quad + |T(\pi_2)| |\min\{1, \frac{p_t}{\mathcal{P}_{\mathbf{X}}(\pi_1)}\} - \min\{1, \frac{p_t}{\mathcal{P}_{\mathbf{X}}(\pi_2)}\}| \\ &\leq |S(\pi_1) - S(\pi_2)| + |T(\pi_1) - T(\pi_2)| \\ &\quad + (|S(\pi_2)| + |T(\pi_2)|) |\min\{1, \frac{p_t}{\mathcal{P}_{\mathbf{X}}(\pi_1)}\} - \min\{1, \frac{p_t}{\mathcal{P}_{\mathbf{X}}(\pi_2)}\}|. \end{aligned}$$

Using the following bounds

$$\begin{aligned} |S(\pi_1) - S(\pi_2)| &\leq (2M_1 + 2M_2 + L) \|\pi_1 - \pi_2\|_{\infty}, \\ |T(\pi_1) - T(\pi_2)| &\leq L \|\pi_1 - \pi_2\|_{\infty}, \\ |S(\pi_2)| &\leq M_1 + M_2, \\ |T(\pi_2)| &\leq M_1 + M_2, \\ |\min\{1, \frac{p_t}{\mathcal{P}_{\mathbf{X}}(\pi_1)}\} - \min\{1, \frac{p_t}{\mathcal{P}_{\mathbf{X}}(\pi_2)}\}| &= |\frac{p_t}{\max\{p_t, \mathcal{P}_{\mathbf{X}}(\pi_1)\}} - \frac{p_t}{\max\{p_t, \mathcal{P}_{\mathbf{X}}(\pi_2)\}}| \\ &\leq \frac{1}{p_t} |\mathcal{P}_{\mathbf{X}}(\pi_1) - \mathcal{P}_{\mathbf{X}}(\pi_2)| \leq \frac{1}{p_t} \|\pi_1 - \pi_2\|_{\infty}, \end{aligned}$$

yields $|S_{p_t}(\pi_1) - S_{p_t}(\pi_2)| \leq 2[(M_1 + M_2 + L) + \frac{1}{p_t}(M_1 + M_2)] \|\pi_1 - \pi_2\|_{\infty}$. Similarly, we also have $|S_{n,p_t}^{\tau,\delta}(\pi_1) - S_{n,p_t}^{\tau,\delta}(\pi_2)| \leq 2[(M_1 + M_2 + L) + \frac{1}{p_t}(M_1 + M_2)] \|\pi_1 - \pi_2\|_{\infty}$. ■

In the same sense as Lemma 2, using Lemma 3 we obtain the following bound for the policy functional class under a capacity constraint p_t .

Lemma 4. Under Assumption 3, for any $\{\mathbf{X}_i\}_{i=1}^n \in \mathcal{X}^{\otimes n}$ and $\epsilon > 0$, it satisfies

$$\Pr \left(\sup_{\pi \in \Pi} |S_{n,p_t}^{\tau,\delta}(\pi) - S_{p_t}(\pi)| \leq \epsilon \right) \geq 1 - \mathcal{N} \exp \left(-\frac{n\epsilon^2}{32(d_{\max}^2 + 1)(M_1 + M_2)^2} \right),$$

where $\mathcal{N} := \mathcal{N} \left(\Pi, \frac{\epsilon}{8[(M_1 + M_2 + L) + \frac{1}{p_t}(M_1 + M_2)]} \right)$ represents the covering number on the policy functional class Π with radius $\frac{\epsilon}{8[(M_1 + M_2 + L) + \frac{1}{p_t}(M_1 + M_2)]}$.

Finally, we can derive the capacity-constrained policy regret bound.

Theorem 5. By Assumption 3, for any small $\epsilon > 0$, the policy regret under the capacity constraint p_t is bounded by $\mathcal{R}(\hat{\pi}_n^{p_t}) \leq 2 \left(\frac{\alpha_\tau}{n^{\zeta_\tau}} + \frac{\alpha_\delta}{n^{\zeta_\delta}} \right) + 2\epsilon$ with probability at least $1 - \mathcal{N} \exp \left(-\frac{n\epsilon^2}{32(d_{\max}^2+1)(M_1+M_2)^2} \right)$, where $\mathcal{N} := \mathcal{N} \left(\Pi, \frac{\epsilon}{8[(M_1+M_2+L)+\frac{1}{p_t}(M_1+M_2)]} \right)$ indicates the covering number on the functional class Π with radius $\frac{\epsilon}{8[(M_1+M_2+L)+\frac{1}{p_t}(M_1+M_2)]}$, and d_{\max} is the maximal node degree in the graph \mathcal{G} .

Proof. Consider an arbitrary policy $\tilde{\pi} \in \Pi$, we have the following utility difference

$$\begin{aligned} S_{p_t}(\tilde{\pi}) - S_{p_t}(\hat{\pi}_n^{p_t}) &\leq \underbrace{S_{n,p_t}^{\tau,\delta}(\tilde{\pi}) - \hat{S}_{n,p_t}^{\tau,\delta}(\tilde{\pi}) - S_{n,p_t}^{\tau,\delta}(\hat{\pi}_n^{p_t}) + \hat{S}_{n,p_t}^{\tau,\delta}(\hat{\pi}_n^{p_t})}_{(1)} \\ &\quad \underbrace{S_{p_t}(\tilde{\pi}) - S_{n,p_t}^{\tau,\delta}(\tilde{\pi}) + S_{n,p_t}^{\tau,\delta}(\hat{\pi}_n^{p_t}) - S_{p_t}(\hat{\pi}_n^{p_t})}_{(2)}. \end{aligned}$$

Using the fact that $\forall \pi \in \Pi$, $|2\pi(\mathbf{X}_i) \min\{1, \frac{p_t}{\mathcal{P}_{\mathbf{X}}(\pi)}\} - 1| \leq 1$, it is easy to see (1) $\leq 2 \left(\frac{\alpha_\tau}{n^{\zeta_\tau}} + \frac{\alpha_\delta}{n^{\zeta_\delta}} \right)$. Furthermore,

$$(2) \leq |S_{n,p_t}^{\tau,\delta}(\tilde{\pi}) - S_{p_t}(\tilde{\pi})| + |S_{n,p_t}^{\tau,\delta}(\hat{\pi}_n^{p_t}) - S_{p_t}(\hat{\pi}_n^{p_t})| \leq 2 \sup_{\pi \in \Pi} |S_{n,p_t}^{\tau,\delta}(\pi) - S_{p_t}(\pi)|.$$

In summary, via Lemma 4 it yields the statement. \blacksquare

F.3 Error Bound of Causal Estimators under Interference

In this section we will give a heuristic explanation why the causal estimators are difficult to obtain under interference. First, with abuse of notation, we consider the following linear model with deterministic outcome

$$\mu_\star(\mathbf{X}_i, \mathbf{X}, \mathbf{T}, \mathcal{G}) = T_i \tau_\star(\mathbf{X}_i) + \alpha_1 \sum_{j \in \mathcal{N}_i} T_j \tau_\star(\mathbf{X}_j) + \alpha_2 \sum_{k \in \mathcal{N}_i^{(2)}} T_k \tau_\star(\mathbf{X}_k) \quad (18)$$

by setting $Y_i(T_i = 0) = 0$, $\alpha = 1$ and letting $\alpha_1 = \frac{1}{|\mathcal{N}_i|}$, $\alpha_2 = \frac{1}{|\mathcal{N}_i^{(2)}|}$, where τ_\star stands for the ground truth individual treatment response which is bounded by $\|\tau_\star\|_\infty \leq M$.

One motivation for employing localized graph convolution network, such as GraphSAGE, is that the surrogate model of a 2-layer GraphSAGE can recover the linear model, especially, when $\mathbf{T} = \mathbf{1}$. To be more specific, consider the following form of a 2-layer GraphSAGE

$$\begin{aligned} \mathbf{X}_i^{(1)} &= \text{ReLU}(\mathbf{X}_i + \sum_{j \in \mathcal{N}_i} \mathbf{X}_j \mathbf{W}^{(1)}) \\ \mathbf{X}_i^{(2)} &= \text{ReLU}(\mathbf{X}_i^{(1)} + \sum_{j \in \mathcal{N}_i} \mathbf{X}_j^{(1)} \mathbf{W}^{(2)}) \\ &= \text{ReLU}[\text{ReLU}(\mathbf{X}_i + \sum_{j \in \mathcal{N}_i} \mathbf{X}_j \mathbf{W}^{(1)}) + \sum_{j \in \mathcal{N}_i} \text{ReLU}(\mathbf{X}_j + \sum_{k \in \mathcal{N}_i^{(2)}} \mathbf{X}_k \mathbf{W}^{(1)}) \mathbf{W}^{(2)}]. \end{aligned}$$

A prediction from it reads $o(\mathbf{X}_i) = \mathbf{X}_i^{(2)\top} \mathbf{v}$, where \mathbf{v} is a vector mapping the second hidden layer to the outcome prediction. In a *surrogate model*⁶, where an identity mapping replaces the ReLU activation function, the model returns the outcome prediction

$$o_{\text{surrogate}}(\mathbf{X}_i) = \mathbf{X}_i^\top \mathbf{v} + \sum_{j \in \mathcal{N}_i} (\mathbf{X}_j \mathbf{W}^{(1)} + \mathbf{X}_j \mathbf{W}^{(2)})^\top \mathbf{v} + \sum_{k \in \mathcal{N}_i^{(2)}} (\mathbf{X}_k \mathbf{W}^{(1)} \mathbf{W}^{(2)})^\top \mathbf{v},$$

which correctly recovers the linear model and the simulation protocol of spillover effects when all units are assigned to treatment. Moreover, according to the universal approximation properties of GNNs [34], μ_\star can be approximated. However, this claim cannot reflect an explicit dependence of estimation error on the graph structure. Hence, motivated by the *surrogate model* and the universal approximation property, we study the following class of functions derived from the universal GNN. Let \mathcal{T} be a class of bounded functions with envelop $M < \infty$ and finite VC-dimension $VC(\mathcal{T}) < \infty$, and let

$$\mathcal{M}_{GNN} := \{\tau_1 + \dots + \tau_{D_{max}}, \tau_i \in \mathcal{T} \cup \{0\}, i = 1, \dots, D_{max}, \|\tau_1 + \dots + \tau_{D_{max}}\|_\infty \leq 3M\}, \quad (19)$$

where D_{max} is related to the maximal degree of the graph, for a 2-layer GNN $D_{max} := 1 + d_{max} + d_{max}^2$. Function from \mathcal{M}_{GNN} takes $(\mathbf{X}_i, \mathbf{X}_{j \in \mathcal{N}_i}, \mathbf{X}_{k \in \mathcal{N}_i^{(2)}})_{i=1}^n$ as input⁷ and returns outcome prediction. The maximal subscript D_{max} serves as a padding, to fit it, the function class \mathcal{T} is extended to $\mathcal{T} \cup \{0\}$. As an example, one can find a function $\mu_{GNN} \in \mathcal{M}_{GNN}$ which approximates $\mu_\star(\mathbf{X}_i, \mathbf{X}, \mathbf{T}, \mathcal{G})$ as

$$\mu_{GNN}(\mathbf{X}_i, \mathbf{X}, \mathbf{T}, \mathcal{G}) = \tau_0(\mathbf{X}_i) + \sum_{j \in \mathcal{N}_i} \tau_j(\mathbf{X}_j) + \sum_{k \in \mathcal{N}_i^{(2)}} \tau_k(\mathbf{X}_k),$$

where $\tau_0, \tau_j, \tau_k \in \mathcal{T}$, for $j \in \mathcal{N}_i, k \in \mathcal{N}_i^{(2)}$. In other words, there exists a function in the class \mathcal{M}_{GNN} which, for every node in the network, only uses the representations of this node, this node's neighbors, and this node's 2-hop neighbors, similar to the surrogate model. Assumptions used in this section are summarized in Assumption 4.

Assumption 4.

(A1) Outcome simulation under interference follows the protocol given in Eq. 18 with $\|\mu_\star\|_\infty \leq 3M$ due to the requirement $\|\tau_\star\|_\infty \leq M$.

(A2) Outcome prediction model is drawn from \mathcal{M}_{GNN} defined in Eq. 19.

(A3) There are no isolated nodes in the network⁸.

Define the best approximation realized by the class \mathcal{M}_{GNN} as

$$\tilde{\mu}_{GNN} := \operatorname{argmin}_{\mu \in \mathcal{M}_{GNN}} \|\mu - \mu_\star\|_\infty,$$

and the approximation error

$$\epsilon_{GNN} := \|\tilde{\mu}_{GNN} - \mu_\star\|_\infty. \quad (20)$$

⁶ The *surrogate models* of graph convolutional networks are first studied in [45] for designing adversarial attacks on GNNs and finding robust nodes.

⁷ Note that, treatment assignments can be combined with the covariates and fed into the function. In the experiments, we fed $T_i \mathbf{X}_i$ into the GNNs, meaning that only covariates of treated units are non-zero.

⁸ This assumption will be used later

Moreover, define the optimal empirical estimator as

$$\hat{\mu}_{GNN} := \operatorname{argmin}_{\mu \in \mathcal{M}_{GNN}} \sum_{i=1}^n \ell(\mu(\mathbf{X}_i, \mathbf{X}, \mathbf{T}, \mathcal{G}), Y_i).$$

Since both $\tilde{\mu}_{GNN}$ and $\hat{\mu}_{GNN}$ belong to the same class \mathcal{M}_{GNN} , it is easy to see

$$\mathbb{E}_n[\ell(\tilde{\mu}_{GNN}(\mathbf{X}_i), Y_i)] \geq \mathbb{E}_n[\ell(\hat{\mu}_{GNN}(\mathbf{X}_i), Y_i)],$$

where we write $\tilde{\mu}_{GNN}(\mathbf{X}_i)$ and $\hat{\mu}_{GNN}(\mathbf{X}_i)$ for the sake of simplicity.

We can decompose the approximation error of the empirical causal estimator using the following fact

$$\begin{aligned} & \mathbb{E}[\ell(\hat{\mu}_{GNN}(\mathbf{X}_i), Y_i) - \ell(\mu_\star(\mathbf{X}_i), Y_i)] \\ &= \mathbb{E}\mathbb{E}_{Y_i}[\hat{\mu}_{GNN}^2(\mathbf{X}_i) - 2Y_i\hat{\mu}_{GNN}(\mathbf{X}_i) + 2Y_i\mu_\star(\mathbf{X}_i) - \mu_\star^2(\mathbf{X}_i)] \\ &= \mathbb{E}[\hat{\mu}_{GNN}^2(\mathbf{X}_i) - 2\mu_\star(\mathbf{X}_i)\hat{\mu}_{GNN}(\mathbf{X}_i) + \mu_\star^2(\mathbf{X}_i)] \\ &= \mathbb{E}[(\hat{\mu}_{GNN}(\mathbf{X}_i) - \mu_\star(\mathbf{X}_i))^2]. \end{aligned}$$

It then yields

$$\begin{aligned} \mathbb{E}[(\hat{\mu}_{GNN}(\mathbf{X}_i) - \mu_\star(\mathbf{X}_i))^2] &= \mathbb{E}[\ell(\hat{\mu}_{GNN}(\mathbf{X}_i), Y_i) - \ell(\mu_\star(\mathbf{X}_i), Y_i)] \\ &\leq \mathbb{E}[\ell(\hat{\mu}_{GNN}(\mathbf{X}_i), Y_i) - \ell(\mu_\star(\mathbf{X}_i), Y_i)] \\ &\quad - \mathbb{E}_n[\ell(\hat{\mu}_{GNN}(\mathbf{X}_i), Y_i)] + \mathbb{E}_n[\ell(\tilde{\mu}_{GNN}(\mathbf{X}_i), Y_i)] \\ &= \underbrace{(\mathbb{E} - \mathbb{E}_n)[\ell(\hat{\mu}_{GNN}(\mathbf{X}_i), Y_i) - \ell(\mu_\star(\mathbf{X}_i), Y_i)]}_{(I)} \\ &\quad + \underbrace{\mathbb{E}_n[\ell(\tilde{\mu}_{GNN}(\mathbf{X}_i), Y_i) - \ell(\mu_\star(\mathbf{X}_i), Y_i)]}_{(II)}. \end{aligned}$$

The second term (II) can be bounded by applying the Bernstein inequality. The following inequality holds with probability at least $1 - e^{-\gamma}$

$$\begin{aligned} (II) &\leq \mathbb{E}[\ell(\tilde{\mu}_{GNN}(\mathbf{X}_i), Y_i) - \ell(\mu_\star(\mathbf{X}_i), Y_i)] + \sqrt{\frac{2C_\ell^2 \|\tilde{\mu}_{GNN} - \mu_\star\|_\infty^2 \gamma}{n}} \\ &\quad + \frac{2C_\ell \|\tilde{\mu}_{GNN} - \mu_\star\|_\infty \gamma}{3n} \\ &= \mathbb{E}[(\tilde{\mu}_{GNN}(\mathbf{X}_i) - \mu_\star(\mathbf{X}_i))^2] + \sqrt{\frac{2C_\ell^2 \epsilon_{GNN}^2 \gamma}{n}} + \frac{2C_\ell \epsilon_{GNN} \gamma}{3n} \\ &\leq \epsilon_{GNN}^2 + \epsilon_{GNN} \sqrt{\frac{2C_\ell^2 \gamma}{n}} + \frac{4C_\ell M \gamma}{n} \end{aligned} \tag{21}$$

using the facts $\|\ell(\tilde{\mu}_{GNN}(\mathbf{X}_i), Y_i) - \ell(\mu_\star(\mathbf{X}_i), Y_i)\|_\infty \leq C_\ell \|\tilde{\mu}_{GNN} - \mu_\star\|_\infty$ and $\epsilon_{GNN} := \|\tilde{\mu}_{GNN} - \mu_\star\|_\infty \leq 6M$ where C_ℓ represents the finite Lipschitz constant of loss function.

Furthermore, the first term (I), the maximal deviation between empirical and true means, can be bounded using the standard symmetrization method (see Theorem 2.1 in [4]). Consider a class of functions \mathcal{F} , for any $f \in \mathcal{F}$, assume that $\|f\|_\infty \leq F$ and $\mathbb{V}[f] \leq V$. Then for every $\gamma > 0$, with probability at least $1 - e^{-\gamma}$

$$\sup_{f \in \mathcal{F}} (\mathbb{E}[f] - \mathbb{E}_n[f]) \leq \inf_{\alpha > 0} \left(2(1 + \alpha) \mathcal{R}_n \mathcal{F} + \sqrt{\frac{2V\gamma}{n}} + 2F \left(\frac{1}{3} + \frac{1}{\alpha} \right) \frac{\gamma}{n} \right),$$

where $\mathcal{R}_n \mathcal{F}$ indicates the Rademacher complexity of \mathcal{F} . Hence, it gives

$$(I) \leq 4\mathcal{R}_n\{\ell(\mu) - \ell(\mu_\star) : \mu \in \mathcal{M}_{GNN}\} + 6\sqrt{\frac{2C_\ell^2 M^2 \gamma}{n}} + \frac{16C_\ell M \gamma}{n} \quad (22)$$

by setting $\alpha = 1$ and using $\|\ell(\mu(\mathbf{X}_i), Y_i) - \ell(\mu_\star(\mathbf{X}_i), Y_i)\|_\infty \leq C_\ell \|\mu - \mu_\star\|_\infty \leq 6C_\ell M$, $\mathbb{V}[\ell(\mu(\mathbf{X}_i), Y_i) - \ell(\mu_\star(\mathbf{X}_i), Y_i)] \leq \mathbb{E}[(\ell(\mu(\mathbf{X}_i), Y_i) - \ell(\mu_\star(\mathbf{X}_i), Y_i))^2] \leq 36C_\ell^2 M^2$ for any $\mu \in \mathcal{M}_{GNN}$. Moreover, the Rademacher complexity term is defined as

$$\begin{aligned} & \mathcal{R}_n\{\ell(\mu) - \ell(\mu_\star) : \mu \in \mathcal{M}_{GNN}\} \\ &:= \mathbb{E}_\sigma \left[\sup_{\mu \in \mathcal{M}_{GNN}} \left| \frac{1}{n} \sum_{i=1}^n \sigma_i (\ell(\mu(\mathbf{X}_i), Y_i) - \ell(\mu_\star(\mathbf{X}_i), Y_i)) \right| \middle| \mathbf{X}, \mathbf{T}, \mathcal{G} \right] \\ &\leq C_\ell \underbrace{\mathbb{E}_\sigma \left[\sup_{\mu \in \mathcal{M}_{GNN}} \left| \frac{1}{n} \sum_{i=1}^n \sigma_i (\mu(\mathbf{X}_i) - \mu_\star(\mathbf{X}_i)) \right| \middle| \mathbf{X}, \mathbf{T}, \mathcal{G} \right]}_{(\#)}, \end{aligned} \quad (23)$$

where $\{\sigma_i\}_{i=1}^n$ are Rademacher random variables. Before using the covering number arguments to further bound the Rademacher complexity term we introduce the following lemmas.

Lemma 5 (Theorem 29.6 in [8]). *Let $\mathcal{F}_1, \dots, \mathcal{F}_k$ be classes of real functions on \mathbb{R}^d . For n arbitrary fixed points $z_1^n = (z_1, \dots, z_n)$ in \mathbb{R}^d , define the sets $\mathcal{F}_1(z_1^n), \dots, \mathcal{F}_k(z_1^n)$ by $\mathcal{F}_j(z_1^n) = \{f_j(z_1), \dots, f_j(z_n) : f_j \in \mathcal{F}_j\}$, $j = 1, \dots, k$. Also introduce $\mathcal{F} = \{f_1 + \dots + f_k : f_j \in \mathcal{F}_j, j = 1, \dots, k\}$. Then for every $\epsilon > 0$ and z_1^n ,*

$$\mathcal{N}_1(\epsilon, \mathcal{F}(z_1^n)) \leq \prod_{j=1}^k \mathcal{N}_1(\epsilon/k, \mathcal{F}_j(z_1^n)). \quad (24)$$

Lemma 6. *Let $\mathcal{F}_1, \dots, \mathcal{F}_k$ be classes of bounded real functions on \mathbb{R}^d with envelop F and finite VC-dimension $v < \infty$, for $3 \leq k \leq K$. Also introduce $\mathcal{F} = \{f_1 + \dots + f_k, f_j \in \mathcal{F}_j, j = 1, \dots, k\}$ and let $\mathcal{F}(z_1^n) = \{f(z_1), \dots, f(z_n), f \in \mathcal{F}\}$ for arbitrary fixed points z_1^n in \mathbb{R}^d . Then we have the following bound*

$$\mathbb{E}_\sigma \left[\sup_{f \in \mathcal{F}} \left| \frac{1}{n} \sum_{i=1}^n \sigma_i f(z_i) \right| \right] \leq C_F \sqrt{\frac{kv \ln k}{n}}, \quad (25)$$

where C_F is a constant which depends only on the envelop.

Proof. According to the Theorem 5.22 in [43], the Rademacher complexity term is bounded as

$$\mathbb{E}_\sigma \left[\sup_{f \in \mathcal{F}} \left| \frac{1}{n} \sum_{i=1}^n \sigma_i f(z_i) \right| \right] \leq \underbrace{\frac{32}{\sqrt{n}} \int_0^{2F} \sqrt{\ln \mathcal{N}_1(\epsilon, \mathcal{F}(z_1^n))} d\epsilon}_{(\star)}.$$

Using Lemma 5 and $\mathcal{N}_1(\epsilon, \mathcal{F}) \leq \mathcal{N}_2(\epsilon, \mathcal{F})$, it gives

$$(\star) \leq \frac{32}{\sqrt{n}} \int_0^{2F} \sqrt{\sum_{j=1}^k \ln \mathcal{N}_2(\epsilon/k, \mathcal{F}_j(z_1^n))}.$$

Moreover, a uniform entropy bound for the covering number is given by the Theorem 2.6.7 in [41]. A small modification gives

$$\mathcal{N}_2(\epsilon, \mathcal{F}_j(z_1^n)) \leq C(v+1)(16e)^{(v+1)}(k/\epsilon)^{2v}, j = 1, \dots, k,$$

where C is a universal constant. Furthermore, following the same technique used by Eq. A.6 in [23], we obtain

$$\begin{aligned} (\star) &\leq \frac{32}{\sqrt{n}} \sqrt{k} \int_0^{2F} \sqrt{\ln C + \ln(v+1) + (v+1) \ln(16e) + 2v \ln k - 2v \ln \epsilon} d\epsilon \\ &\stackrel{(1)}{\leq} \frac{32}{\sqrt{n}} \sqrt{kv} \int_0^{2F} \sqrt{\ln C + \ln 2 + \ln(16e) + 2 \ln k - 2 \ln \epsilon} d\epsilon \\ &\stackrel{(2)}{\leq} \frac{32}{\sqrt{n}} \sqrt{kv \ln k} \int_0^{2F} \sqrt{\ln C + \ln 2 + \ln(16e) + 2 - 2 \ln \epsilon / \ln K} d\epsilon := C_F \sqrt{\frac{kv \ln k}{n}}, \end{aligned}$$

where (1) uses the fact that usually v is large enough and (2) is due to the condition $3 \leq k \leq K$. ■

Now, we can further bound the term $\mathcal{R}_n\{\ell(\mu) - \ell(\mu_\star) : \mu \in \mathcal{M}_{GNN}\}$ after Eq. 23. Note that

$$\begin{aligned} (\#) &= C_\ell \mathbb{E}_\sigma \left[\sup_{f \in \mathcal{M}_{GNN}} \left| \frac{1}{n} \sum_{i=1}^n \sigma_i [(\tau_0(\mathbf{X}_i) - T_i \tau_\star(\mathbf{X}_i)) + \sum_{j \in \mathcal{N}_i} (\tau_j(\mathbf{X}_j) - T_j \tau_\star(\mathbf{X}_j))] \right. \right. \\ &\quad \left. \left. + \sum_{k \in \mathcal{N}_i^{(2)}} (\tau_k(\mathbf{X}_k) - T_k \tau_\star(\mathbf{X}_k)) \right] \right]. \end{aligned} \quad (26)$$

Define a new constant for each node $D_i := 1 + |\mathcal{N}_i| + |\mathcal{N}_i^{(2)}|$, $i = 1, \dots, n$. According to (A3) in Assumption 4, we have $D_i \geq 3$. Also introduce a new class of function $\Omega := \{\mathcal{T} \pm \tau_\star\}$. Note that class Ω has the same VC-dimension as \mathcal{T} , i.e., $VC(\Omega) = VC(\mathcal{T})$, and $\|\omega\|_\infty \leq 2M$ for any $\omega \in \Omega$. Recall the definition of $D_{max} := 1 + d_{max} + d_{max}^2$. By decomposing the node subscript i into groups with the same D_i , Eq. 26 can be further written as

$$\begin{aligned} (\#) &= C_\ell \mathbb{E}_\sigma \left[\sup_{f \in \mathcal{M}_{GNN}} \left| \frac{1}{n} \sum_{i=1}^n \sigma_i \sum_{l=1}^{D_i} \omega_l(\mathbf{X}_i, \mathbf{X}, \mathbf{T}, \mathcal{G}) \right| \right] \quad \omega_l \in \Omega \\ &= C_\ell \mathbb{E}_\sigma \left[\sup_{f \in \mathcal{M}_{GNN}} \left| \sum_{k=3}^{D_{max}} \frac{1}{n} \sum_{i: D_i=k} \sigma_i \sum_{l=1}^k \omega_l(\mathbf{X}_i, \mathbf{X}, \mathbf{T}, \mathcal{G}) \right| \right] \\ &\stackrel{(1)}{\leq} C_\ell \sum_{k=3}^{D_{max}} \mathbb{E}_\sigma \left[\sup_{f \in \mathcal{M}_{GNN}} \left| \frac{1}{n} \sum_{i: D_i=k} \sigma_i \sum_{l=1}^k \omega_l(\mathbf{X}_i, \mathbf{X}, \mathbf{T}, \mathcal{G}) \right| \right] \\ &\stackrel{(2)}{\leq} C_\ell C_F \sum_{k=3}^{D_{max}} \frac{1}{n} \sqrt{|i : D_i = k| k VC(\mathcal{T}) \ln k} \leq C_\ell C_F \sum_{k=3}^{D_{max}} \sqrt{\frac{k VC(\mathcal{T}) \ln k}{n}} \\ &\leq C_\ell C_F \sqrt{\frac{D_{max}^3 VC(\mathcal{T}) \ln D_{max}}{n}}, \end{aligned}$$

where (1) uses the triangle inequality and (2) uses Lemma 6. Hence, the (I) term is bounded by

$$(I) \leq 4C_\ell C_F \sqrt{\frac{D_{max}^3 VC(\mathcal{T}) \ln D_{max}}{n}} + 6\sqrt{\frac{2C_\ell^2 M^2 \gamma}{n}} + \frac{16C_\ell M \gamma}{n}.$$

By combining (I) and (II) we have the following theorem.

Theorem 6. *Suppose Assumption 4 holds. Let $\hat{\mu}_{GNN}$ be the optimal causal estimator obtained by minimizing an empirical loss function using the data $\{\mathbf{X}_i, \mathbf{X}, \mathbf{T}, \mathcal{G}\}_{i=1}^n$. Suppose that the loss function has a finite Lipschitz constant C_ℓ and $\hat{\mu}_{GNN}$ is restricted to \mathcal{M}_{GNN} . Then with probability at least $1 - 2e^{-\gamma}$, the causal estimator under interference has an error bound*

$$\mathbb{E}[(\hat{\mu}_{GNN}(\mathbf{X}_i) - \mu_\star(\mathbf{X}_i))^2] \leq 4C_\ell C_F \sqrt{\frac{D_{max}^3 VC(\mathcal{T}) \ln D_{max}}{n}} + 6\sqrt{\frac{2C_\ell^2 M^2 \gamma}{n}} \quad (27)$$

$$+ \epsilon_{GNN} \sqrt{\frac{2C_\ell^2 \gamma}{n}} + \frac{20C_\ell M \gamma}{n} + \epsilon_{GNN}^2, \quad (28)$$

where ϵ_{GNN} is defined in Eq. 20.

Keeping only the leading term with D_{max} , under network interference, the causal estimator has an error bound $\mathcal{O}(\sqrt{\frac{D_{max}^3 \ln D_{max}}{n}})$. It indicates that an accurate causal estimator is difficult to obtain under large network interference. Recall that the prediction outcome from the GNN causal estimator is actually the superposition of individual treatment effect and spillover effect. Hence, it is expected that, similarly, the individual treatment effect becomes more and more difficult to recover under more substantial network interference. This intuitive expectation can be observed in the following experimental results in Table 18. We observe that the error of individual treatment effect estimator increases from $k = 1$ to $k = 4$.

Table 18: ϵ_{PEHE} on the semi-synthetic Wave1 data with $p = 0.1$, $\alpha = 0.5$, and $k = 1, 2, 4$. To fit the theoretical analysis, exposure level is not fed into the model.

	$k = 1$	$k = 2$	$k = 4$
GraphSAGE	0.048	0.129	0.152

Chapter 7

Conclusion

In this dissertation, we investigated machine learning on relational data from the perspectives of cognition, quantum computing, and causal inference. Examples of relational data considered here are graphs with undirected edges, e.g., social networks, and directed graphs with labeled edges such as knowledge graphs. Mainly, we studied the connections between relational databases, or knowledge graphs, and cognitive memory functions. We developed quantum machine learning algorithms to accelerate the inference on knowledge graphs. Besides, we developed causal inference algorithms on networks with interfered causal effects.

In Chapter 2, we discussed the technical realizations and mathematical models for declarative memories. Semantic and episodic knowledge graphs were considered as the technical realizations of semantic and episodic memory, respectively. They were modeled by decomposing the corresponding adjacency tensors. After tensor decomposition, the obtained latent representations of subjects, predicates, and objects can capture the global relational patterns of a semantic knowledge graph for memorization and implicit knowledge inference. Modeling episodic knowledge graphs returns additionally representations for timestamps, which are essential for reconstructing episodic tensors and generalizing on time-dependent facts. We have demonstrated the importance of the high dimensionality of timestamp representations for generalization and, especially, for the memorization of sparse episodic tensors. Besides, we have shown that semantic memory can be derived from episodic memory via marginalization over the time dimension, realizing a transfer from time-dependent facts to static facts. As future works, we will investigate deeper relationships between knowledge graphs and cognitive memory functions in the context of visual understanding by incorporating sensory and working memories. More advanced

methods of modeling the episodic knowledge graphs for future events prediction are also future works.

In Chapter 3, we discussed a previous cognitive architecture for associative memory, the holographic reduced representation, and proposed a method to improve the memory capacity. We have demonstrated that the memory interference caused by the superposition of different association pairs in one single memory trace can be dramatically reduced by elementwise sampling initial random vectors from heavy-tailed distributions, e.g., the Cauchy distribution. We mathematically proved how an improved quasi-orthogonality could increase the associative memory capacity by deriving the distribution function of pairwise angles between random vectors. We showed how the holographic reduced representation could be adjusted and applied to semantic knowledge graphs. The resulting representations, or the holistic representations, implement a distributed storage of semantic pairs and can be used for predicting implicit links by incorporating a simple neural network. Analogously, the performance of holistic representations in the generalization task can be improved by introducing the Cauchy initialization. The derived distribution function of pairwise angles between quasi-orthogonal random vectors might find unexpected applications in random projection and dimension reduction, serving as future works.

To resolve the slow inference issue caused by the increasing number of semantic triples and entities, in Chapters 4 and 5, we proposed two quantum algorithms to accelerate the reasoning on semantic knowledge graphs. The first quantum approach in Chapter 4 introduced quantum representations for entities by employing parametric quantum circuits and encoding the representations to the amplitudes of quantum states. Hence, this quantum Ansatz is a learning-based approach and can heuristically realize a quadratic speedup when inferring unobserved triples. Recall that variational quantum circuits can be viewed as multi-layer linear neural networks without nonlinear activation functions. According to the superior performance on the dataset with rather simple relational patterns, future work could be introducing nonlinear activations into the quantum circuit Ansätze to model more complicated relational patterns. In Chapter 5, the second quantum approach designed a quantum counterpart of the classical tensor singular value decomposition algorithm, making it a sampling-based quantum algorithm. For knowledge inference, it realizes an exponential acceleration with dimensions of knowledge graphs. The theoretical contributions, under what conditions the tensor decomposition of the subsampled and rescaled tensor can well approximate the original tensor, deserve more considerations in the future work.

In the last chapter of this dissertation, we studied causal effects estimation on networks in the framework of Neyman-Rubin causal inference. It is a nontrivial task since the relational nature of the network makes the treatment effects of individuals not independent anymore, which is known as network interference or spillover effect. Hence, we have proposed GNN-based causal estimators for causal inference under network interference and shown their superior performance on synthetic and real datasets. After obtaining optimal causal estimators, we defined a novel utility function and policy network to maximize the average welfare on the network and confirmed the importance of employing an accurate causal estimator for the intervention policy improvement under network interference. As independent theoretical contributions, we derived heuristic error bounds for GNN-based causal estimators and the regret bound of the policy network. The graph-dependency of the error bonds could be refined using more advanced concentration inequalities of dependent variables. Besides, we will consider causal inference under network interference with only partially observed or dynamic graph structures.

This dissertation has only discussed a small portion of machine learning on relational data. For instance, the ongoing works are using the neural point process for future events prediction from observed temporal knowledge graphs, defining representations on non-Euclidean manifolds to learn the complex, or hierarchical, relational structures in knowledge graphs, understanding cognitive memory functions with knowledge graphs and perceptions. Therefore, leaning with relational data is an exciting and challenging research area, and there remains a lot to be explored.

Bibliography

- [1] Odd Aalen, Ornulf Borgan, and Hakon Gjessing. *Survival and event history analysis: a process point of view*. Springer Science & Business Media, 2008.
- [2] Dimitris Achlioptas and Frank McSherry. Fast computation of low-rank matrix approximations. *Journal of the ACM (JACM)*, 54(2):9, 2007.
- [3] Esma Aïmeur, Gilles Brassard, and Sébastien Gambs. Quantum speed-up for unsupervised learning. *Machine Learning*, 90(2):261–287, 2013.
- [4] Peter M Aronow, Cyrus Samii, et al. Estimating average causal effects under general interference, with application to a social network experiment. *The Annals of Applied Statistics*, 11(4):1912–1947, 2017.
- [5] Susan Athey and Stefan Wager. Efficient policy learning. *arXiv preprint arXiv:1702.02896*, 2017.
- [6] James Atwood and Don Towsley. Diffusion-convolutional neural networks. In *Advances in neural information processing systems*, pages 1993–2001, 2016.
- [7] Peter C Austin. An introduction to propensity score methods for reducing the effects of confounding in observational studies. *Multivariate behavioral research*, 46(3):399–424, 2011.
- [8] Alan Baddeley. Cognitive psychology and human memory. *Trends in neurosciences*, 11(4):176–181, 1988.
- [9] Stephan Baier, Yunpu Ma, and Volker Tresp. Improving visual relationship detection using semantic modeling of scene descriptions. In *International Semantic Web Conference*, pages 53–68. Springer, 2017.

- [10] Stephan Baier, Yunpu Ma, and Volker Tresp. Improving information extraction from images with learned semantic models. In *IJCAI*, 2018.
- [11] Tim Berners-Lee, James Hendler, and Ora Lassila. The semantic web. *Scientific american*, 284(5):34–43, 2001.
- [12] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250, 2008.
- [13] Elizabeth Boschee, Jennifer Lautenschlager, Sean O’Brien, Steve Shellman, James Starz, and Michael Ward. Icews coded event data. *Harvard Dataverse*, 12, 2015.
- [14] Jake Bowers, Mark M. Fredrickson, and Costas Panagopoulos. Reasoning about interference between units: A general framework. *Political Analysis*, 21:97–124, 2013.
- [15] Gilles Brassard, Peter Hoyer, Michele Mosca, and Alain Tapp. Quantum amplitude amplification and estimation. *Contemporary Mathematics*, 305:53–74, 2002.
- [16] T Tony Cai and Tiefeng Jiang. Phase transition in limiting distributions of coherence of high-dimensional random matrices. *Journal of Multivariate Analysis*, 107:24–39, 2012.
- [17] Tony Cai, Jianqing Fan, and Tiefeng Jiang. Distributions of angles in random packing on spheres. *The Journal of Machine Learning Research*, 14(1):1837–1864, 2013.
- [18] Kim Chantala and Joyce Tabor. National longitudinal study of adolescent health: Strategies to perform a design-based analysis using the add health data. 1999.
- [19] Jie Chen and Yousef Saad. On the tensor svd and the optimal low rank orthogonal approximation of tensors. *SIAM Journal on Matrix Analysis and Applications*, 30(4):1709–1734, 2009.
- [20] David Roxbee Cox and David R Cox. *Planning of experiments*, volume 20. Wiley New York, 1958.
- [21] Shib Sankar Dasgupta, Swayambhu Nath Ray, and Partha Talukdar. Hyte: Hyperplane-based temporally aware knowledge graph embedding. In *Proceedings*

- of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2001–2011, 2018.
- [22] Lieven De Lathauwer, Bart De Moor, and Joos Vandewalle. A multilinear singular value decomposition. *SIAM journal on Matrix Analysis and Applications*, 21(4):1253–1278, 2000.
- [23] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in neural information processing systems*, pages 3844–3852, 2016.
- [24] Persi Diaconis and David Freedman. Asymptotics of graphical projection pursuit. *The annals of statistics*, pages 793–815, 1984.
- [25] Jeffrey L Elman. Finding structure in time. *Cognitive science*, 14(2):179–211, 1990.
- [26] Edward Farhi, Jeffrey Goldstone, and Sam Gutmann. A quantum approximate optimization algorithm. *arXiv preprint arXiv:1411.4028*, 2014.
- [27] Edward Farhi and Hartmut Neven. Classification with quantum neural networks on near term processors. *arXiv preprint arXiv:1802.06002*, 2018.
- [28] David Ferrucci, Eric Brown, Jennifer Chu-Carroll, James Fan, David Gondek, Aditya A Kalyanpur, Adam Lally, J William Murdock, Eric Nyberg, John Prager, et al. Building watson: An overview of the deepqa project. *AI magazine*, 31(3):59–79, 2010.
- [29] Laura Forastiere, Edoardo M Airoidi, and Fabrizia Mealli. Identification and estimation of treatment and interference effects in observational studies on networks. *arXiv preprint arXiv:1609.06245*, 2016.
- [30] Demeter Gábor. Associative holographic memories. *IBM Journal of Research and Development*, 13(2):156–159, 1969.
- [31] Alberto García-Durán, Sebastijan Dumančić, and Mathias Niepert. Learning sequence encoders for temporal knowledge graph completion. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4816–4821. Association for Computational Linguistics, 2018.
- [32] Michael S Gazzaniga. *The cognitive neurosciences*. MIT press, 2009.

- [33] Vittorio Giovannetti, Seth Lloyd, and Lorenzo Maccone. Quantum random access memory. *Physical review letters*, 100(16):160501, 2008.
- [34] Boris Vladimirovich Gnedenko and Andreï Kolmogorov. *Limit distributions for sums of independent random variables*.
- [35] Marco Gori, Gabriele Monfardini, and Franco Scarselli. A new model for learning in graph domains. In *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, volume 2, pages 729–734. IEEE, 2005.
- [36] Daniel L Greenberg and Mieke Verfaellie. Interdependence of episodic and semantic memory: evidence from neuropsychology. *Journal of the International Neuropsychological society*, 16(5):748–753, 2010.
- [37] Arthur Gretton, Olivier Bousquet, Alex Smola, and Bernhard Schölkopf. Measuring statistical dependence with hilbert-schmidt norms. In *International conference on algorithmic learning theory*, pages 63–77. Springer, 2005.
- [38] Lov K Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, pages 212–219, 1996.
- [39] Gian Giacomo Guerreschi and Mikhail Smelyanskiy. Practical optimization for hybrid quantum-classical algorithms. *arXiv preprint arXiv:1701.01450*, 2017.
- [40] Will Hamilton, Payal Bajaj, Marinka Zitnik, Dan Jurafsky, and Jure Leskovec. Embedding logical queries on knowledge graphs. In *Advances in Neural Information Processing Systems*, pages 2026–2037, 2018.
- [41] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems*, pages 1024–1034, 2017.
- [42] Zhen Han, Yuyi Wang, Yunpu Ma, Stephan Günnemann, and Volker Tresp. The graph hawkes network for reasoning on temporal knowledge graphs. *arXiv preprint arXiv:2003.13432*, 2020.
- [43] Aram W Harrow, Avinatan Hassidim, and Seth Lloyd. Quantum algorithm for linear systems of equations. *Physical review letters*, 103(15):150502, 2009.

- [44] Richard A Harshman et al. Foundations of the parafac procedure: Models and conditions for an "explanatory" multimodal factor analysis. 1970.
- [45] Alan G Hawkes. Spectra of some self-exciting and mutually exciting point processes. *Biometrika*, 58(1):83–90, 1971.
- [46] Mikael Henaff, Joan Bruna, and Yann LeCun. Deep convolutional networks on graph-structured data. *arXiv preprint arXiv:1506.05163*, 2015.
- [47] Marcel Hildebrandt, Jorge Andres Quintero Serna, Yunpu Ma, Martin Ringsquandl, Mitchell Joblin, and Volker Tresp. Reasoning on knowledge graphs with debate dynamics. In *AAAI*, 2020.
- [48] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.
- [49] John J Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the national academy of sciences*, 79(8):2554–2558, 1982.
- [50] Michael G. Hudgens and M. Elizabeth Halloran. Toward causal inference with interference. 103(482), 2008.
- [51] Svante Janson. Large deviations for sums of partly dependent random variables. *Random Structures & Algorithms*, 24(3):234–248, 2004.
- [52] Fredrik Johansson, Uri Shalit, and David Sontag. Learning representations for counterfactual inference. In *International conference on machine learning*, pages 3020–3029, 2016.
- [53] Iordanis Kerenidis, Jonas Landman, Alessandro Luongo, and Anupam Prakash. q-means: A quantum algorithm for unsupervised machine learning. In *Advances in Neural Information Processing Systems*, pages 4136–4146, 2019.
- [54] Iordanis Kerenidis and Anupam Prakash. Quantum Recommendation Systems. In Christos H. Papadimitriou, editor, *8th Innovations in Theoretical Computer Science Conference (ITCS 2017)*, volume 67 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 49:1–49:21. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2017.

- [55] Markus Kiefer and Friedemann Pulvermüller. Conceptual representations in mind and brain: theoretical developments, current evidence and future directions. *cortex*, 48(7):805–825, 2012.
- [56] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations (ICLR)*, 2017.
- [57] Alexei Y. Kitaev. Quantum measurements and the abelian stabilizer problem. *Electronic Colloquium on Computational Complexity (ECCC)*, 3, 1995.
- [58] Graham Klyne, Jeremy J Carroll, and B McBride. Resource description framework (rdf): concepts and abstract syntax, 2004. *February*. URL: <http://www.w3.org/TR/2004/REC-rdf-concepts-20040210>, 2009.
- [59] Daphne Koller, Nir Friedman, Sašo Džeroski, Charles Sutton, Andrew McCallum, Avi Pfeffer, Pieter Abbeel, Ming-Fai Wong, David Heckerman, Chris Meek, et al. *Introduction to statistical relational learning*. MIT press, 2007.
- [60] Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. Neural architectures for named entity recognition. In *HLT-NAACL*, 2016.
- [61] Ora Lassila, Ralph R Swick, et al. Resource description framework (rdf) model and syntax specification. 1998.
- [62] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.
- [63] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.
- [64] Kalev Leetaru and Philip A Schrodtt. Gdelt: Global data on events, location, and tone, 1979–2012. In *ISA annual convention*, volume 2, pages 1–49. Citeseer, 2013.
- [65] Jure Leskovec, Lada A Adamic, and Bernardo A Huberman. The dynamics of viral marketing. *ACM Transactions on the Web (TWEB)*, 1(1):5, 2007.

- [66] Lan Liu and Michael G Hudgens. Large sample randomization inference of causal effects in the presence of interference. *Journal of the american statistical association*, 109(505):288–301, 2014.
- [67] Seth Lloyd, Masoud Mohseni, and Patrick Rebentrost. Quantum principal component analysis. *Nature Physics*, 10(9):631, 2014.
- [68] Yunpu Ma, Marcel Hildebrandt, Volker Tresp, and Stephan Baier. Holistic representations for memorization and inference. In *UAI*, pages 403–413, 2018.
- [69] Yunpu Ma, Volker Tresp, and Erik A Daxberger. Embedding models for episodic knowledge graphs. *Journal of Web Semantics*, 59:100490, 2019.
- [70] Yunpu Ma, Volker Tresp, Liming Zhao, and Yuyi Wang. Variational quantum circuit model for knowledge graph embedding. *Advanced Quantum Technologies*, 2(7-8):1800078, 2019.
- [71] Yunpu Ma, Yuyi Wang, and Volker Tresp. Causal inference under networked interference. *arXiv preprint arXiv:2002.08506*, 2020.
- [72] Yunpu Ma, Yuyi Wang, and Volker Tresp. Quantum machine learning algorithm for knowledge graphs. *arXiv preprint arXiv:2001.01077*, 2020.
- [73] Farzaneh Mahdisoltani, Joanna Biega, and Fabian M Suchanek. Yago3: A knowledge base from multilingual wikipedias. 2013.
- [74] Marc Maier, Brian Taylor, Huseyin Oktay, and David Jensen. Learning causal models of relational domains. In *Twenty-Fourth AAAI Conference on Artificial Intelligence*, 2010.
- [75] Jarrod R McClean, Jonathan Romero, Ryan Babbush, and Alán Aspuru-Guzik. The theory of variational hybrid quantum-classical algorithms. *New Journal of Physics*, 18(2):023023, 2016.
- [76] Hongyuan Mei and Jason M Eisner. The neural hawkes process: A neurally self-modulating multivariate point process. In *Advances in Neural Information Processing Systems*, pages 6754–6764, 2017.
- [77] Marvin Minsky. A framework for representing knowledge. 1974.

- [78] Kosuke Mitarai, Makoto Negoro, Masahiro Kitagawa, and Keisuke Fujii. Quantum circuit learning. *Physical Review A*, 98(3):032309, 2018.
- [79] Stephen L Morgan and Christopher Winship. *Counterfactuals and causal inference*. Cambridge University Press, 2015.
- [80] Christopher Morris, Martin Ritzert, Matthias Fey, William L. Hamilton, Jan Eric Lenssen, Gaurav Rattan, and Martin Grohe. Weisfeiler and leman go neural: Higher-order graph neural networks. In *AAAI*, 2019.
- [81] Robb J Muirhead. *Aspects of multivariate statistical theory*, volume 197. John Wiley & Sons, 2009.
- [82] David Nadeau and Satoshi Sekine. A survey of named entity recognition and classification. *Linguisticae Investigationes*, 30(1):3–26, 2007.
- [83] Maximilian Nickel, Kevin Murphy, Volker Tresp, and Evgeniy Gabrilovich. A review of relational machine learning for knowledge graphs. *Proceedings of the IEEE*, 104(1):11–33, 2015.
- [84] Maximilian Nickel, Lorenzo Rosasco, and Tomaso Poggio. Holographic embeddings of knowledge graphs. In *Thirtieth Aaai conference on artificial intelligence*, 2016.
- [85] Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. A three-way model for collective learning on multi-relational data. In *Icml*, volume 11, pages 809–816, 2011.
- [86] Elizabeth L Ogburn, Oleg Sofrygin, Ivan Diaz, and Mark J van der Laan. Causal inference for social network data. *arXiv preprint arXiv:1705.08527*, 2017.
- [87] Elizabeth L Ogburn, Tyler J VanderWeele, et al. Vaccines, contagion, and social networks. *The Annals of Applied Statistics*, 11(2):919–948, 2017.
- [88] Judea Pearl et al. Causal inference in statistics: An overview. *Statistics surveys*, 3:96–146, 2009.
- [89] Tony A Plate. Holographic reduced representations. *IEEE Transactions on Neural networks*, 6(3):623–641, 1995.
- [90] Anupam Prakash. *Quantum algorithms for linear algebra and machine learning*. PhD thesis, UC Berkeley, 2014.

- [91] Rodrigo Quian Quiroga, Itzhak Fried, and Christof Koch. Brain cells for grandmother. *Scientific American*, 308(2):30–35, 2013.
- [92] Patrick Rebentrost, Masoud Mohseni, and Seth Lloyd. Quantum support vector machine for big data classification. *Physical review letters*, 113(13):130503, 2014.
- [93] Paul R Rosenbaum et al. *Design of observational studies*, volume 10. Springer, 2010.
- [94] Paul R Rosenbaum and Donald B Rubin. The central role of the propensity score in observational studies for causal effects. *Biometrika*, 70(1):41–55, 1983.
- [95] Donald B Rubin. Estimating causal effects of treatments in randomized and nonrandomized studies. *Journal of educational Psychology*, 66(5):688, 1974.
- [96] Donald B Rubin. Randomization analysis of experimental data: The fisher randomization test comment. *Journal of the American statistical association*, 75(371):591–593, 1980.
- [97] Donald B Rubin. Causal inference using potential outcomes: Design, modeling, decisions. *Journal of the American Statistical Association*, 100(469):322–331, 2005.
- [98] Stuart Russell and Peter Norvig. Artificial intelligence: a modern approach. 2002.
- [99] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1):61–80, 2008.
- [100] Jürgen Schmidhuber. Deep learning in neural networks: An overview. *Neural networks*, 61:85–117, 2015.
- [101] Maria Schuld, Alex Bocharov, Krysta M Svore, and Nathan Wiebe. Circuit-centric quantum classifiers. *Physical Review A*, 101(3):032308, 2020.
- [102] Uri Shalit, Fredrik D Johansson, and David Sontag. Estimating individual treatment effect: generalization bounds and algorithms. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 3076–3085. JMLR. org, 2017.
- [103] Amit Singhal. Introducing the knowledge graph: things, not strings. *Official google blog*, 16, 2012.

- [104] John F Sowa. Semantic networks. 1987.
- [105] Fabian M Suchanek, Gjergji Kasneci, and Gerhard Weikum. Yago: a core of semantic knowledge. In *Proceedings of the 16th international conference on World Wide Web*, pages 697–706, 2007.
- [106] Lubos Takac and Michal Zabovsky. Data analysis in public social networks. In *International Scientific Conference and International Workshop Present Day Trends of Innovations*, volume 1, 2012.
- [107] Ewin Tang. A quantum-inspired classical algorithm for recommendation systems. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, pages 217–228, 2019.
- [108] Eric J Tchetgen Tchetgen and Tyler J VanderWeele. On causal inference in the presence of interference. *Statistical methods in medical research*, 21(1):55–75, 2012.
- [109] Timothy J Teyler and Pascal DiScenna. The hippocampal memory indexing theory. *Behavioral neuroscience*, 100(2):147, 1986.
- [110] Panos Toulis and Edward Kao. Estimation of causal peer influence effects. In *International conference on machine learning*, pages 1489–1497, 2013.
- [111] Volker Tresp, Cristóbal Esteban, Yinchong Yang, Stephan Baier, and Denis Krompaß. Learning with memory embeddings. *arXiv preprint arXiv:1511.07972*, 2015.
- [112] Volker Tresp, Yunpu Ma, and Stephan Baier. Tensor memories, 2017.
- [113] Volker Tresp, Yunpu Ma, Stephan Baier, and Yinchong Yang. Embedding learning for declarative memories. In *European Semantic Web Conference*, pages 202–216. Springer, 2017.
- [114] Volker Tresp, Sahand Sharifzadeh, Dario Konopatzki, and Yunpu Ma. The tensor brain: Semantic decoding for perception and memory. *arXiv preprint arXiv:2001.11027*, 2020.
- [115] Rakshit Trivedi, Hanjun Dai, Yichen Wang, and Le Song. Know-evolve: Deep temporal reasoning for dynamic knowledge graphs. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 3462–3471. JMLR. org, 2017.

- [116] Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. Complex embeddings for simple link prediction. *International Conference on Machine Learning (ICML)*, 2016.
- [117] Ledyard R Tucker. Some mathematical notes on three-mode factor analysis. *Psychometrika*, 31(3):279–311, 1966.
- [118] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. *International Conference on Learning Representations*, 2018.
- [119] Davide Viviano. Policy targeting under network interference. *arXiv preprint arXiv:1906.10258*, 2019.
- [120] Denny Vrandečić and Markus Krötzsch. Wikidata: a free collaborative knowledge-base. *Communications of the ACM*, 57(10):78–85, 2014.
- [121] Philip R Westlake. The possibilities of neural holographic processes within the brain. *Kybernetik*, 7(4):129–153, 1970.
- [122] Nathan Wiebe, Daniel Braun, and Seth Lloyd. Quantum algorithm for data fitting. *Physical review letters*, 109(5):050505, 2012.
- [123] Nathan Wiebe, Ashish Kapoor, and Krysta Marie Svore. Quantum algorithms for nearest-neighbor methods for supervised and unsupervised learning. *Quantum Information and Computation*, 15:316–356, 2015.
- [124] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S. Yu. A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems*, 2020.
- [125] Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. Embedding entities and relations for learning and inference in knowledge bases. *arXiv preprint arXiv:1412.6575*, 2014.
- [126] Zhitao Ying, Jiaxuan You, Christopher Morris, Xiang Ren, Will Hamilton, and Jure Leskovec. Hierarchical graph representation learning with differentiable pooling. In *Advances in neural information processing systems*, pages 4800–4810, 2018.

- [127] Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. Ernie: Enhanced language representation with informative entities. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1441–1451, 2019.